



Red Hat Certificate System 8.0 Managing Smart Cards with the Enterprise Security Client

Managing personal certificates with the Enterprise Security Client for
Red Hat Certificate System 7.3
Edition 8.0.7

Landmann

Red Hat Certificate System 8.0 Managing Smart Cards with the Enterprise Security Client

Managing personal certificates with the Enterprise Security Client for
Red Hat Certificate System 7.3
Edition 8.0.7

Landmann
rlandmann@redhat.com

Legal Notice

Copyright © 2009 Red Hat, Inc..

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This guide is for regular users of Certificate System subsystems. It explains how to manage personal certificates and keys using the Enterprise Security Client, a simple interface which formats and manages smart cards.

Table of Contents

About This Guide	4
1. What Is in This Guide	4
2. Additional Reading	4
3. Examples and Formatting	5
3.1. Formatting for Examples and Commands	5
3.2. Tool Locations	5
3.3. Guide Formatting	5
4. Giving Feedback	6
5. Document History	6
Chapter 1. Introduction to the Enterprise Security Client	8
1.1. Red Hat Enterprise Linux, Single Sign-On, and Authentication	8
1.2. Red Hat Certificate System and the Enterprise Security Client	9
1.3. The Enterprise Security Client and the Windows Cryptographic Service Provider	10
1.4. About the Mac TokenD Component	11
Chapter 2. Installing the Enterprise Security Client	12
2.1. Supported Platforms for the Client	12
2.2. Supported Smart Cards	12
2.3. Installing and Uninstalling the Enterprise Security Client on Red Hat Enterprise Linux	12
2.3.1. Installing the Client	12
2.3.2. Uninstalling on Red Hat Enterprise Linux	13
2.4. Installing and Uninstalling on Windows	14
2.4.1. Installing the Client	14
2.4.2. Uninstalling the Client	18
2.5. Installing and Uninstalling the Enterprise Security Client on Mac OS X	19
2.5.1. Installing the Client	19
2.5.2. Uninstalling the Client	21
Chapter 3. Using the Enterprise Security Client	23
3.1. Tray Icons for the Enterprise Security Client	23
3.2. Launching Enterprise Security Client	23
3.2.1. Opening the Enterprise Security Client on Red Hat Enterprise Linux	24
3.2.2. Opening the Enterprise Security Client on Microsoft Windows	24
3.2.3. Opening the Enterprise Security Client on Mac OS X	24
3.3. Configuring Phone Home	24
3.3.1. About Phone Home Profiles	25
3.3.2. Setting Global Phone Home Information	25
3.3.3. Adding Phone Home Information to a Token Manually	26
3.3.4. Configuring the TPS to Use Phone Home	27
3.4. Setting up Users to Be Enrolled	28
3.5. Enrolling a Smart Card Automatically	28
3.6. Managing Smart Cards	31
3.6.1. Formatting the Smart Card	32
3.6.2. Resetting a Smart Card Password	32
3.6.3. Viewing Certificates	33
3.6.4. Importing CA Certificates	34
3.6.5. Adding Exceptions for Servers	36
3.6.6. Enrolling Smart Cards	38
3.6.7. Re-Enrolling Tokens	39
3.7. Verifying That the Mac TokenD Is Working Properly	39
3.8. Diagnosing Problems	40

3.8.1. Errors	42
3.8.2. Events	43
Chapter 4. Using Security Officer Mode	44
4.1. Enabling Security Officer Mode	44
4.2. Enrolling a New Security Officer	46
4.3. Using Security Officers to Manage Users	48
4.3.1. Enrolling a New User	48
4.3.2. Performing Other Security Officer Tasks	50
4.3.3. Formatting an Existing Security Officer Smart Card	51
Chapter 5. Using Keys on Smart Cards for Web and Mail Clients	53
5.1. Setting up Browsers to Support SSL for Tokens	53
5.2. Using the Certificates on Tokens for Mail Clients	54
5.3. Setting up Mail and Browser Clients on Mac OS X	55
Chapter 6. Configuring the Enterprise Security Client	57
6.1. Overview of Enterprise Security Client Configuration	57
6.1.1. About the Preferences Configuration Files	57
6.1.2. About the XUL and JavaScript Files in the Enterprise Security Client	63
6.1.3. Enterprise Security Client File Locations	63
6.2. Configuring SSL Connections with the TPS	65
6.3. Using Shared Security Databases	66
6.4. Customizing the Smart Card Enrollment User Interface	66
6.5. Disabling LDAP Authentication for Token Operations	71

About This Guide

The Enterprise Security Client is a simple user interface which formats and manages smart cards. This guide is intended for everyday users of Certificate System, who use the Enterprise Security Client to manage their smart cards. Certificate System agents should read the *Certificate System Agent's Guide* for information on how to perform agent tasks, such as handling certificate requests and revoking certificates.

Before reading this guide, be familiar with the following concepts:

- Public-key cryptography and the Secure Sockets Layer (SSL) protocol
- Intranet, extranet, Internet security, and the role of digital certificates in a secure enterprise
- LDAP and Red Hat Directory Server

1. What Is in This Guide

This guide contains the following chapters:

- [Chapter 1, *Introduction to the Enterprise Security Client*](#) provides an introduction to the Certificate System.
- [Section 2.1, "Supported Platforms for the Client"](#) provides information about supported platforms for the Enterprise Security Client.
- [Chapter 2, *Installing the Enterprise Security Client*](#) provides information on how to install and uninstall the Enterprise Security Client on the supported platforms.
- [Chapter 3, *Using the Enterprise Security Client*](#) provides instructions on using the Enterprise Security Client for token enrollment, formatting, and password reset operations.
- [Chapter 5, *Using Keys on Smart Cards for Web and Mail Clients*](#) describes how to use the Enterprise Security Client keys for SSL and S/MIME authentication.
- [Chapter 6, *Configuring the Enterprise Security Client*](#) provides information on configuring the Enterprise Security Client.

2. Additional Reading

This paper covers information on managing smart cards in Certificate System, as well as the functionality for the Enterprise Security Client, which handles smart cards.

Some very basic information on using other end user web services for the Certificate System CA and RA systems is covered in the [Using End User Services](#). For basic certificate management, that's all many users need to know. *Managing Smart Cards with the Enterprise Security Client* and the *End User's Guide*, together, are both for end users of Red Hat Certificate System.

For more information on the basic concepts of certificates, public key infrastructure, and Certificate System itself, see the [Certificate System Deployment Guide](#).

More detailed information about the concepts behind public key cryptography, as well as a more detailed overview of the Certificate System subsystems and how Certificate System manages certificates and smart cards, is available in the [Certificate System Administrator's Guide](#). This is also the guide for administrators to manage the Certificate System server. Installation is covered in the [Certificate System Installation Guide](#).

The [Certificate System Agent's Guide](#) covers how agents can approve and reject certificate requests and manage user certificates through other Certificate System subsystems, such as the Online Certificate Status Responder (which checks the revocation status) and the Data Recovery Manager

(which recovers the certificate information if a token or a certificate is lost).

The latest information about Red Hat Certificate System, including current release notes and other updates, is always available at the Certificate System documentation page, <http://www.redhat.com/docs/manuals/cert-system/>.

3. Examples and Formatting

3.1. Formatting for Examples and Commands

All of the examples for Red Hat Certificate System commands, file locations, and other usage are given for Red Hat Enterprise Linux 5 (32-bit) systems. Be certain to use the appropriate commands and files for your platform.

Example 1. Example Command

To start the Red Hat Certificate System:

```
service pki-ca start
```

3.2. Tool Locations

All of the tools for Red Hat Certificate System are located in the `/usr/bin` directory. These tools can be run from any location without specifying the tool location.

3.3. Guide Formatting

Certain words are represented in different fonts, styles, and weights. Different character formatting is used to indicate the function or purpose of the phrase being highlighted.

Formatting Style	Purpose
Monospace font Monospace with a background	Monospace is used for commands, package names, files and directory paths, and any text displayed in a prompt. This type of formatting is used for anything entered or returned in a command prompt.
<i>Italicized text</i>	Any text which is italicized is a variable, such as <i>instance_name</i> or <i>hostname</i>. Occasionally, this is also used to emphasize a new term or other phrase.
Bolded text	Most phrases which are in bold are application names, such as Cygwin, or are fields or options in a user interface, such as a User Name Here: field or Save button.

Other formatting styles draw attention to important text.

**NOTE**

A note provides additional information that can help illustrate the behavior of the system or provide more detail for a specific issue.

**IMPORTANT**

Important information is necessary, but possibly unexpected, such as a configuration change that will not persist after a reboot.

**WARNING**

A warning indicates potential data loss, as may happen when tuning hardware for maximum performance.

4. Giving Feedback

If there is any error in this *Enterprise Security Client Guide* or there is any way to improve the documentation, please let us know. Bugs can be filed against the documentation for Red Hat Certificate System through Bugzilla, <http://bugzilla.redhat.com/bugzilla>. Make the bug report as specific as possible, so we can be more effective in correcting any issues:

- Select the Red Hat Certificate System product.
- Set the component to **Doc - enterprise-security-guide**.
- Set the version number to 8.0.
- For errors, give the page number (for the PDF) or URL (for the HTML), and give a succinct description of the problem, such as incorrect procedure or typo.
For enhancements, put in what information needs to be added and why.
- Give a clear title for the bug. For example, "**Incorrect command example for setup script options**" is better than "**Bad example**".

We appreciate receiving any feedback — requests for new sections, corrections, improvements, enhancements, even new ways of delivering the documentation or new styles of docs. You are welcome to contact Red Hat Content Services directly at docs@redhat.com.

5. Document History

Revision 8.0.7	June 1, 2010	Ella Deon Lackey
Adding back Mac support, on a new platform (10.5.8), per Errata RHBA-2010:0448.		
Revision 8.0.6	January 23, 2010	Ella Deon Lackey
Correcting Windows Enterprise Security Client process for proper 64-bit Windows packages, per Errata RHBA-2009:1687.		
Revision 8.0.5	December 21, 2009	Ella Deon Lackey
Updating platform support to include 64-bit Windows platforms, per Errata RHBA-2009:1687.		

Adding note on escaping characters in passwords for LDAP auth, per Bugzilla #523568.
Adding new configuration parameter and section on configuring shared terminals, per Errata RHBA-2009:1687.
Adding note on new error message if connection to TPS format screen times out, per Errata RHBA-2009:1665.

Revision 8.0.4	September 4, 2009	Ella Deon Lackey
-----------------------	--------------------------	-------------------------

Rearranging introduction and adding information on single sign-on.
Fixing directory path to esc-prefs.js file, per Bugzilla #459511.

Revision 8.0.3	August 20, 2009	Ella Deon Lackey
-----------------------	------------------------	-------------------------

Changing screenshot in Windows installation, per Bugzilla #518314.

Revision 8.0.2	August 13, 2009	Ella Deon Lackey
-----------------------	------------------------	-------------------------

Correcting missed or updated edits for Bugzilla #459511 and #510569.

Revision 8.0.1	July 29, 2009	Ella Deon Lackey
-----------------------	----------------------	-------------------------

Tech reviews for each chapter, per Bugzilla #510564, #510566, #510567, #510569, #510571, and #510572.
Edits to the parameters listed for the esc-pref.js file, per Bugzilla #459511.

Revision 8.0.0	July 22, 2009	Ella Deon Lackey
-----------------------	----------------------	-------------------------

Initial draft for Certificate System 8.0 *Enterprise Security Client Guide*.

Chapter 1. Introduction to the Enterprise Security Client

The Enterprise Security Client is a tool for Red Hat Certificate System which simplifies managing smart cards. End users can use security tokens (smart cards) to store user certificates used for applications such as single sign-on access and client authentication. End users are issued the tokens containing certificates and keys required for signing, encryption, and other cryptographic functions.

The Enterprise Security Client is the third part of Certificate System's complete token management system. Two subsystems — the Token Key Service (TKS) and Token Processing System (TPS) — are used to process token-related operations. The Enterprise Security Client is the interface which allows the smart card and user to access the token management system.

After a token is enrolled, applications such as Mozilla Firefox and Thunderbird can be configured to recognize the token and use it for security operations, like client authentication and S/MIME mail. The Enterprise Security Client provides the following capabilities:

- ▶ Supports Global Platform-compliant smart cards like Gemalto 64K V2 and Safenet 300J Java smart cards.
- ▶ Enrolls security tokens so they are recognized by TPS.
- ▶ Maintains the security token, such as re-enrolling a token with TPS.
- ▶ Provides information about the current status of the token or tokens being managed.
- ▶ Supports server-side key generation through the TPS and DRM subsystems so that keys can be archived and recovered on a separate token if a token is lost.

1.1. Red Hat Enterprise Linux, Single Sign-On, and Authentication

Network users frequently have to submit multiple passwords for the various services they use, such as email, web browsing and intranets, and servers on the network. Maintaining multiple passwords, and constantly being prompted to enter them, is a hassle for users and administrators. *Single sign-on* is a configuration which allows administrators to create a single password store so that users can log in once, using a single password, and be authenticated to all network resources.

Red Hat Enterprise Linux 5.3 supports single sign-on for several resources, including logging into workstations and unlocking screensavers, accessing encrypted web pages using Mozilla Firefox, and sending encrypted email using Mozilla Thunderbird.

Single sign-on is both a convenience to users and another layer of security for the server and the network. Single sign-on hinges on secure and effective authentication, and the Enterprise Security Client ties into the public-key infrastructure implemented by Red Hat Certificate System.

One of the cornerstones of establishing a secure network environment is making sure that access is restricted to people who have the right to access the network. If access is allowed, users can *authenticate* to the system, meaning they can verify their identities. One method of verifying an identity is presenting a certificate. A certificate is an electronic document which identifies the entity which presents it.

These certificates can be stored on a smart card. When a user inserts a smart card, the smart card presents the certificates to the system and identifies the user so the user can be authenticated. One of the two authentication methods for Red Hat Enterprise Linux's single sign-on is smart card authentication. (The other is Kerberos-based authentication.)

Single sign-on on Red Hat Enterprise Linux using smart cards goes through three steps:

1. A user inserts a smart card into the card reader. This is detected by the pluggable authentication

modules (PAM) on Red Hat Enterprise Linux.

2. The system maps the certificate to the user entry and then compares the presented certificates on the smart card to the certificates stored in the user entry.
3. If the certificate is successfully validated against the key distribution center (KDC), then the user is allowed to log in.

The Enterprise Security Client manages the smart cards, which is part of administering single sign-on.

1.2. Red Hat Certificate System and the Enterprise Security Client

Red Hat Certificate System creates, manages, renews, and revokes certificates and keys. For managing smart cards, the Certificate System has a token management system to generate keys, create certificate requests, and receive certificates.

Two subsystems — the Token Key Service (TKS) and Token Processing System (TPS) — are used to process token-related operations. The Enterprise Security Client is the interface which allows the smart card and user to access the token management system.

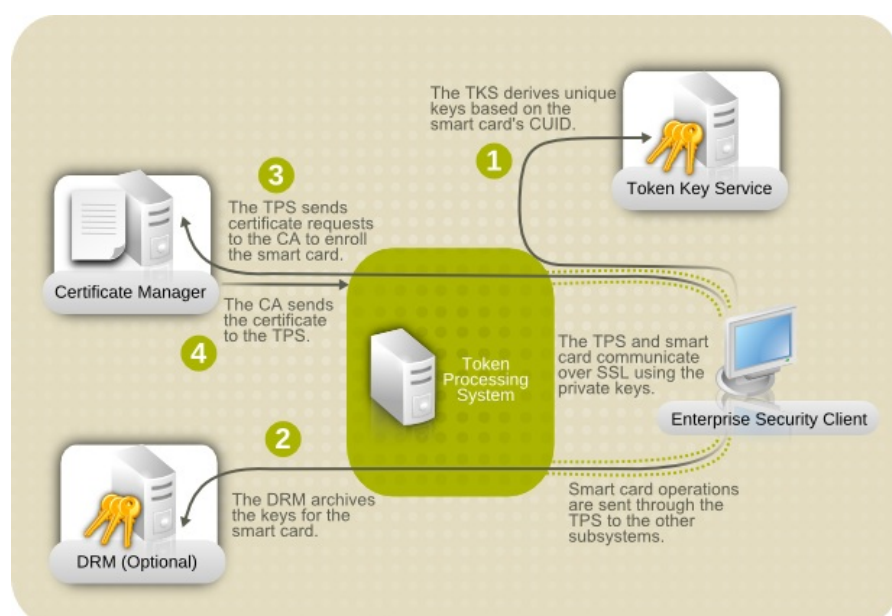


Figure 1.1. How Certificate System Manages Smart Cards

A total of four Certificate System subsystems are involved with managing tokens, two for managing the tokens (TKS and TPS) and two for managing the keys and certificates within the public-key infrastructure (CA and DRM).

- The Token Processing System (TPS) interacts with smart cards to help them generate and store keys and certificates for a specific entity, such as a user or device. Smart card operations go through the TPS and are forwarded to the appropriate subsystem for action, such as the Certificate Authority to generate certificates or the Data Recovery Manager to archive and recover keys.
- The Token Key Service (TKS) generates, or derives, symmetric keys used for communication between the TPS and smart card. Each set of keys generated by the TKS is unique because they are based on the card's unique ID. The keys are formatted on the smart card and are used to encrypt communications, or provide authentication, between the smart card and TPS.
- The Certificate Authority (CA) creates and revokes user certificates stored on the smart card.
- Optionally, the Data Recovery Manager (DRM) archives and recovers keys for the smart card.

The Enterprise Security Client is the conduit through which TPS communicates with each token over a secure HTTP channel (HTTPS), and, through the TPS, with the Certificate System.

To use the tokens, the Token Processing System must be able to recognize and communicate with them. The tokens must first be *enrolled* to populate the tokens with required keys and certificates and add the tokens to the Certificate System. The Enterprise Security Client provides the user interface for end entities to enroll tokens.

1.3. The Enterprise Security Client and the Windows Cryptographic Service Provider

The Microsoft Windows version of the Enterprise Security Client installs a *Windows Cryptographic Service Provider* (CSP) that is compatible with the Certificate System-supported smart cards.

Microsoft Windows supports a software library designed to implement the Microsoft Cryptographic Application Programming Interface (CAPI or CryptoAPI). CAPI allows Windows-based applications, such as the Microsoft Outlook or Internet Explorer, to be developed to perform secure, cryptographic operations. This API provides a layer between these crypto-enabled applications and the details of the cryptographic services provided by the API.

The CAPI interface can be used to create custom CSP libraries. Custom CSP libraries in Certificate System support using smart cards on Windows (enrolled through the Enterprise Security Client) to perform the cryptographic functions requested by Windows applications which access the CAPI interface.

The *CAPI store* is a repository controlled by Windows, and which houses a collection of digital certificates associated with a given CSP. CAPI oversees the certificates, while each CSP controls the cryptographic keys belonging to the certificates.

The Certificate System CSP is designed to provide cryptographic functions on behalf of Windows using our supported smart cards. The Windows CSP performs its requested cryptographic functionality by calling the CoolKey PKCS #11 module.

The Certificate System CSP, which has been signed by Microsoft, enables users to be able to perform common tasks securely:

- Send and receive encrypted and signed emails with Microsoft Outlook.
- Visit SSL-protected websites with Microsoft Internet Explorer.
- Access certain VPN clients using the smart card, which provides secure access to protected networks.
- Log into a Windows server or domain using the smart card, as long as the infrastructure required for smart card login has been properly set up.

The required CSP libraries are automatically installed with the Enterprise Security Client. There are several common situations when a Windows user interacts directly with the CSP.

- When a smart card is enrolled with the Enterprise Security Client, the newly created certificates are automatically inserted into the user's CAPI store.
- When a smart card is formatted, the certificates associated with that card are removed from the CAPI store.
- When a user inserts a properly enrolled card, the certificates on that card are automatically written to the CAPI store. The certificates are then removed when the card is removed from the computer.
- When using applications such as Microsoft Outlook or Microsoft Internet Explorer, the user may be

prompted to enter the smart card's password. This is required when the smart card is asked to perform protected cryptographic operations such as creating digital signatures.

1.4. About the Mac TokenD Component

The Mac Enterprise Security Client ships with a smart card-specific TokenD component which bridges the gap between Certificate System-supported tokens and the Mac CDSA security layer, allowing current OS X applications like Apple Mail and Safari to take advantage of the capabilities of Certificate System tokens:

- ▶ The Mac Keychain Access utility can be used to view the certificates and keys on Certificate System tokens.
- ▶ The Apple Mail client can be used to send and view signed and encrypted emails using Certificate System tokens.
- ▶ The Apple Safari browser can use Certificate System tokens to log onto secure SSL web sites.

Chapter 2. Installing the Enterprise Security Client

The Enterprise Security Client is packaged as a set of installation executables or RPMs and other files that are part of the complete Red Hat Certificate System distribution. These are listed in the installation chapter of the *Certificate System Administrator's Guide*.

2.1. Supported Platforms for the Client

The Enterprise Security Client interface is supported on the following platforms:

- Red Hat Enterprise Linux 5.3 (x86)
- Red Hat Enterprise Linux 5.3 (x86_64)
- Microsoft Windows Vista 32-bit
- Microsoft Windows Vista 64-bit
- Microsoft Windows XP 32-bit
- Microsoft Windows XP 64-bit
- Apple Mac OS X 10.5.8 and higher (Leopard)

2.2. Supported Smart Cards

Enterprise Security Client supports smart cards which are JavaCard 2.1 or higher and Global Platform 2.01-compliant. Certificate System was tested using the following cards:

- Safenet 330J Java smart cards
- Gemalto 64K V2 tokens, both as a smart card and GemPCKey USB form factor key

Smart card testing was conducted using the SCM SCR331 CCID reader.

The only card manager applet supported with Certificate System is the CoolKey applet, one of the packages included and installed with Red Hat Certificate System 8.0.

2.3. Installing and Uninstalling the Enterprise Security Client on Red Hat Enterprise Linux

2.3.1. Installing the Client

The first step in installing the Enterprise Security Client is to download the required packages. The *Certificate System Administrator's Guide* explains how to retrieve these RPMs and other files through the **Red Hat Certificate System 8.0 (AS v.4 for x86)** or **Red Hat Certificate System 8.0 (ES v.4 for x86)** Red Hat Network channels. There are two ways to obtain the packages:

- Downloading an ISO image or packages through the Red Hat Network channel
- Using the Red Hat **yum** utility

The preferred method of obtaining RPMs is using the **yum** command-line utility, as follows:

```
# yum install esc
```

If the **yum** command completes successfully, all of the necessary Enterprise Security Client RPMs will be installed and ready for use.



NOTE

If the **yum** utility was used to install the Enterprise Security Client, there is no need for further installation; the client has already been installed. The following procedure is for installing from a CD image.

1. As **root**, install the Enterprise Security Client packages. On Red Hat Enterprise Linux 5 (32-bit), these packages are already installed, and can be updated using **yum**. For example:

```
yum install esc coolkey
Setting up Install Process
Parsing package install arguments
Resolving Dependencies
--> Running transaction check
---> Package esc.x86_64 0:1.1.0-9.el5 set to be updated
---> Package coolkey.x86_64 0:1.1.0-9.el5 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

=====
===
Package                        Arch      Version           Repository        Size
=====
===
Installing:
  esc                        x86_64    1.1.0-9.el5       base              522 k
Installing for dependencies:
  coolkey                   x86_64    1.1.0-9.el5       base              90
k

Transaction Summary
=====
===
Install      2 Package(s)
Update      0 Package(s)
Remove      0 Package(s)

Total download size: 612 k
Is this ok [y/N]: y
```

The Enterprise Security Client is located in **/usr/lib/esc-1.1.0** on Red Hat Enterprise Linux 32-bit systems and **/usr/lib64/esc-1.1.0** on Red Hat Enterprise Linux 64-bit system. The **esc** shell script is installed in **/usr/bin/esc**. The Enterprise Security Client can be launched by typing **esc** at a command prompt.

The Enterprise Security Client for Linux implements a daemon (**escd**) which runs silently, waiting for a smart card to be inserted. When an unenrolled smart card is inserted, the daemon automatically launches the client UI, and the Enterprise Security Client guides the user through the enrollment process. The client can also be launched manually by selecting **System Settings**, then **Smart Card Manager**, from the **System** menu.

2.3.2. Uninstalling on Red Hat Enterprise Linux

1. Unplug all USB tokens.

2. Stop the Enterprise Security Client.
3. Log in as **root**, and use **rpm -ev** to remove the Enterprise Security Client RPMs in the following order:

**NOTE**

Update the version numbers of the RPM files to match your version.

```
# rpm -ev coolkey
# rpm -ev esc
```

4. Remove any remaining files in the installation directory.

2.4. Installing and Uninstalling on Windows

2.4.1. Installing the Client

The Windows Enterprise Security Client packages are available in the **Downloads** area of Red Hat Network.

For both 32-bit and 64-bit Windows systems, the Windows Enterprise Security Client package is available in the 32-bit channel. This package is **SmartCardManagerSetup-1.0.1-X.win32.i386.exe**.

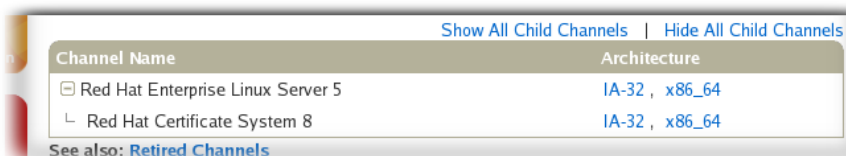
For 64-bit systems, there is an additional package which must be installed, **CoolKeySetup-version.win64.x64.exe**, to use the Enterprise Security Client to manage certificates for email and browser clients.

To install the Enterprise Security Client on Windows:

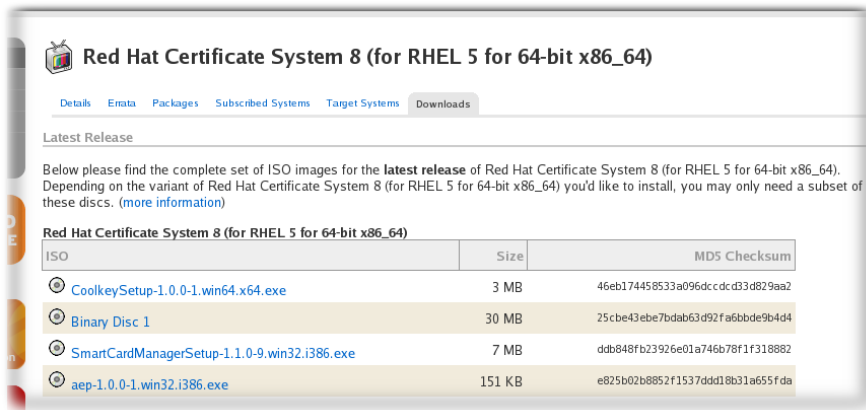
**NOTE**

You may need administrator privileges to install the Enterprise Security Client on the Windows system.

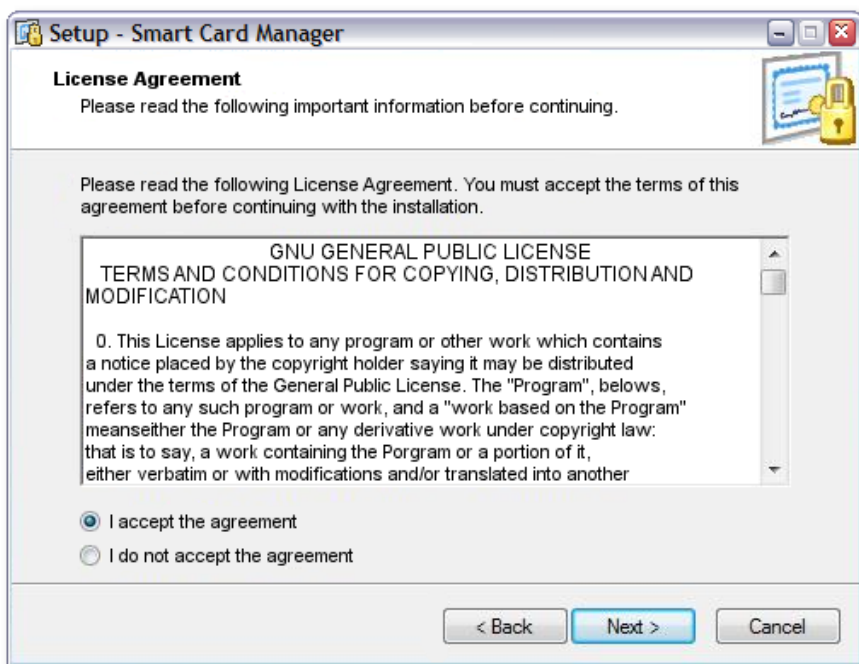
1. In the Red Hat Network page for Certificate System, select the 32-bit or 64-bit channel.



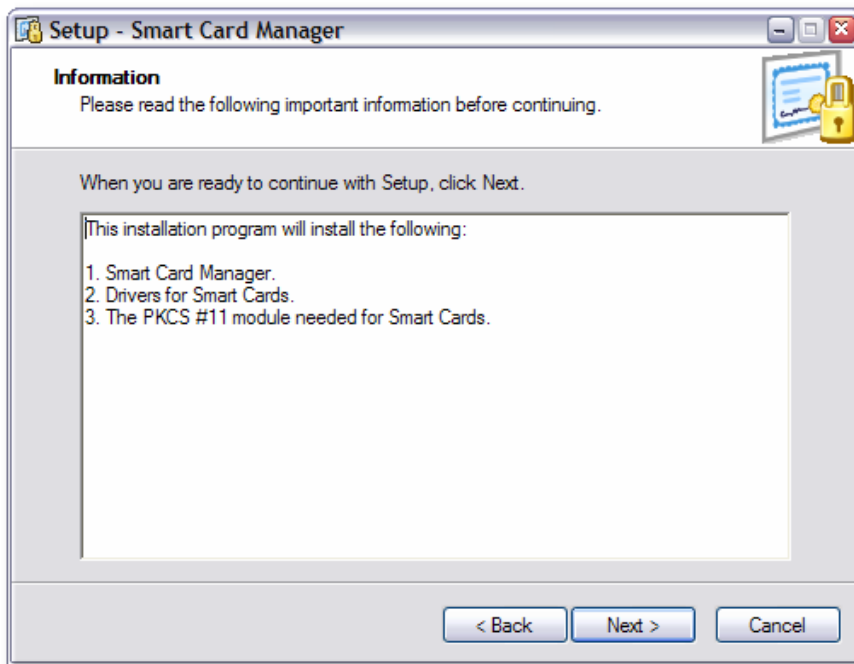
2. Click the **Downloads** tab in the channel.
3. Download the Enterprise Security Client installer and (for 64-bit systems) CoolKey libraries.



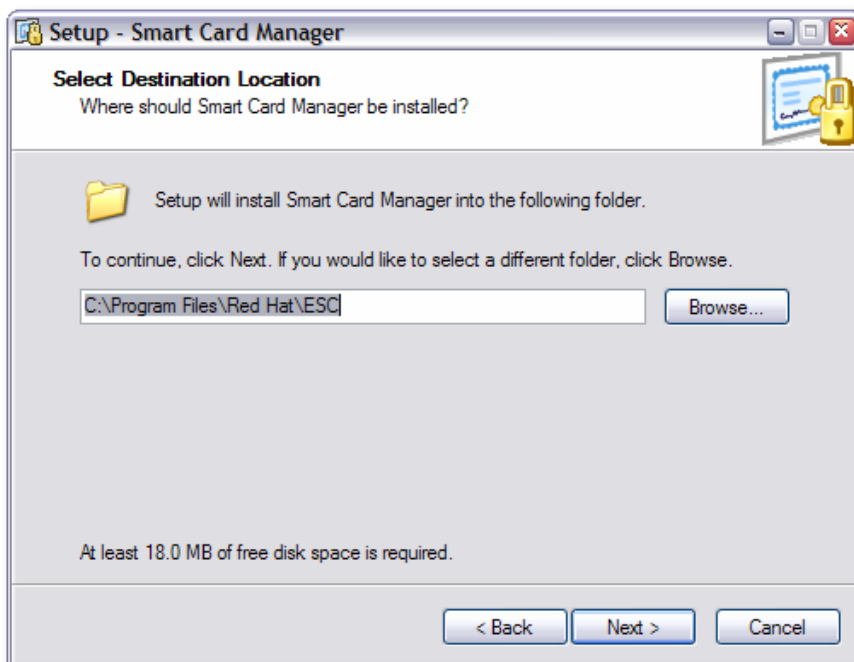
4. Next, double-click the **SmartCardManagerSetup-1.0.1-X.win32.i386.exe** file to launch the Enterprise Security Client installation program.
5. Click **Next** to being going through the installer, and then accept the license agreement.



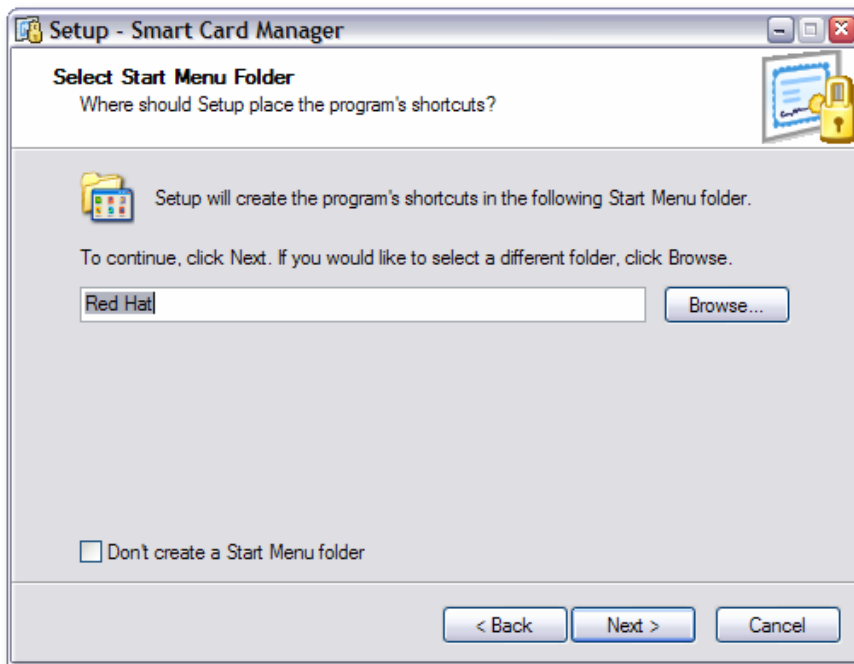
6. The wizard displays the list of packages that will be installed.



7. The wizard prompts for the installation directory for the Enterprise Security Client. The default directory is **C:\Program Files\Red Hat\ESC**.



8. The wizard prompts for the Start Menu directory for the Enterprise Security Client. The default directory is **Red Hat**.



9. Proceed through the Enterprise Security Client installation wizard. Click **Install** to begin installing the Enterprise Security Client components.

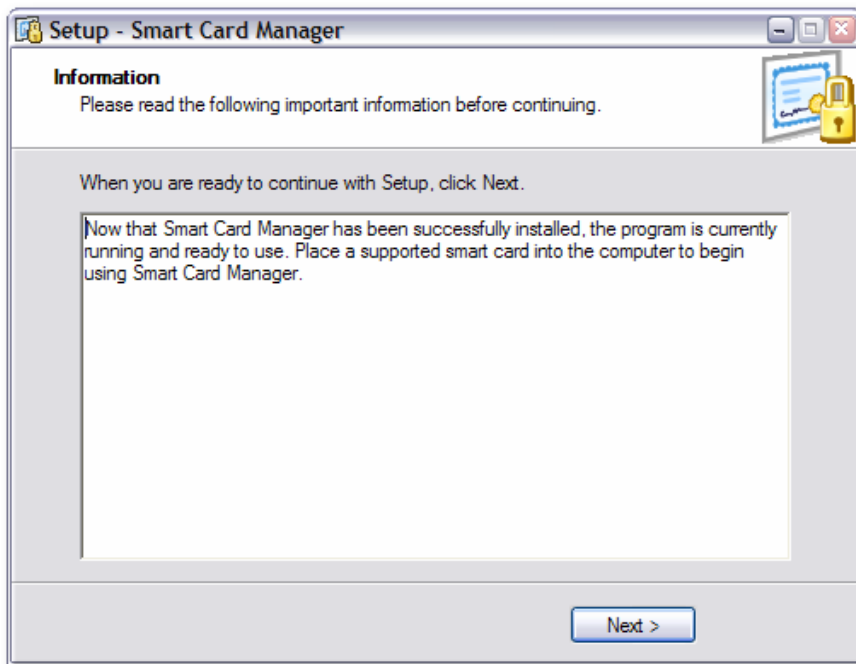


NOTE

The installation process also installs the CoolKey PKCS #11 driver needed for Certificate System-supported keys and automatically installs the Certificate System PKCS #11 module in any Mozilla browsers it can locate. The installer places the Certificate System Cryptographic Service Provider (CSP) on the user's system to allow users to use their smart cards with Microsoft products such as Outlook and Internet Explorer.



10. When the installation has completed, the Enterprise Security Client will prompt the user to insert a token, and can then be launched for immediate use.



11. Click **Finish** to complete the installation. The machine has to be restarted after installing the Enterprise Security Client, so, if possible, select the **Yes** radio button to restart the machine immediately.



12. *For 64-bit systems only.* Last, double-click the **CoolKeySetup-version.win64.x64.exe** file to install the required 64-bit CoolKey libraries.

2.4.2. Uninstalling the Client

1. Unplug all USB tokens.
2. Stop the Enterprise Security Client.
3. Open the **Control Panel**, and click the **Add Remove Programs** icon.
4. In the list of available programs, click **Smart Card Manager**, and click **Change/Remove**.
5. When the uninstallation is complete, remove any remaining files in the installation directory.

2.5. Installing and Uninstalling the Enterprise Security Client on Mac OS X

2.5.1. Installing the Client

The Mac Enterprise Security Client packages are available in the **Downloads** area of Red Hat Network. There are two channels for the packages; Mac clients are available in 32-bit. The Mac Smart Card Manager package is **SmartCardManager.version.dmg**.

To install the Enterprise Security Client on Mac OS X:

1. Download the **SmartCardManager-1.0.1-X.OSX4.darwin.dmg** file from the Red Hat Network channel.
2. Double-click the **SmartCardManager-1.0.1-X.OSX4.darwin.dmg** file to display the Enterprise Security Client Volume.
Inside the Volume are two directories, **ESC.app** and **Coolkey1.14.pkg**. **ESC.app** is the Enterprise Security Client application, and **Coolkey1.14.pkg** is the installer for the token support software, including the TokenD system.
3. Drag the **ESC.app** file to an accessible location (for example, the desktop), to install the Enterprise Security Client.
4. Install the CoolKey package, as follows:
 - a. Double-click the **Coolkey1.14.pkg** file to launch the CoolKey installer, and follow the directions to complete the installation.

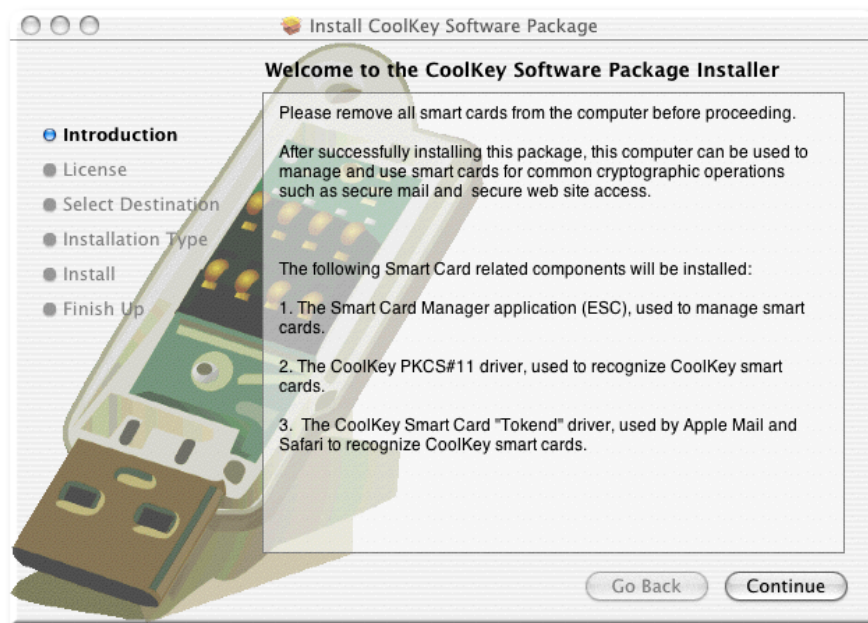


Figure 2.1. Mac Installation Program

- b. Read the Software License Agreement, and click **Continue** if you accept the terms.
- c. Select the installation destination.



Figure 2.2. Specifying the installation destination

- d. Click **Upgrade** (or **Install**, if shown), to begin the installation.

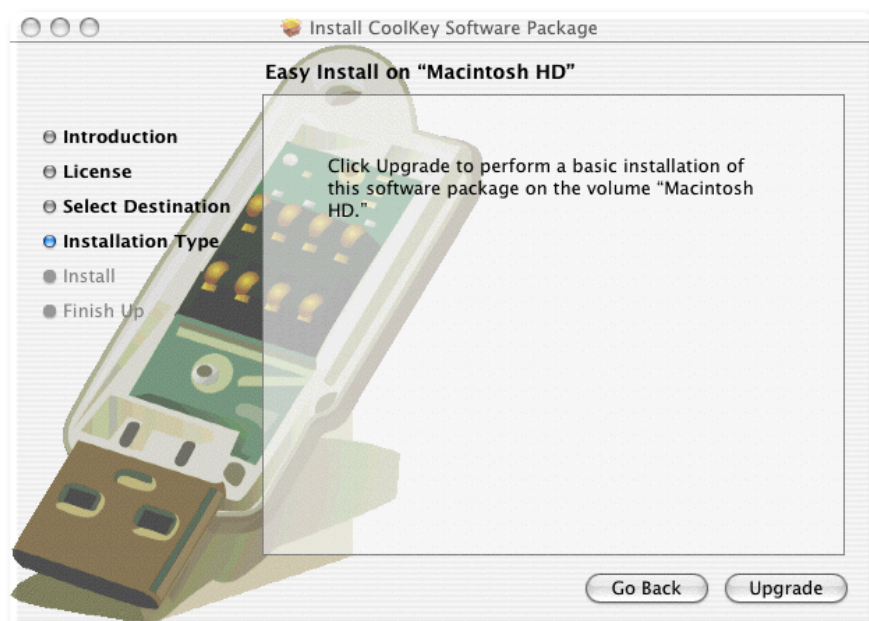


Figure 2.3. Launch Installation

- e. Enter the administrator password, and click **OK** to start the installation.

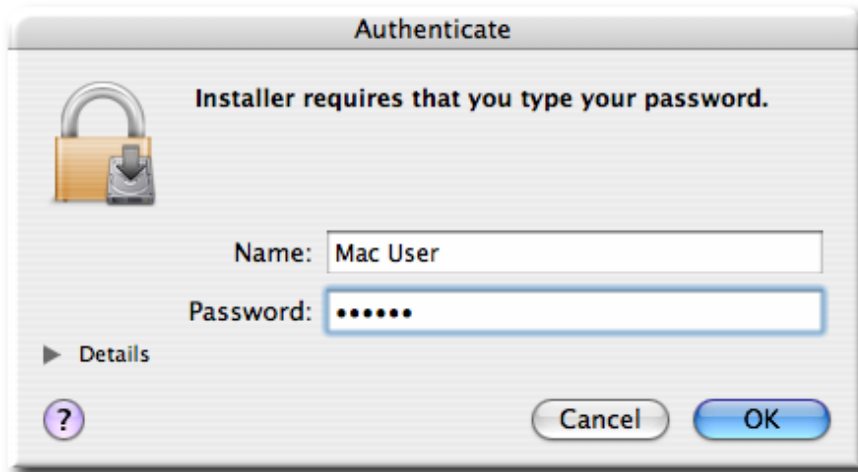


Figure 2.4. Entering the Administrator Password

- f. When the installation is complete, click **C**lose.

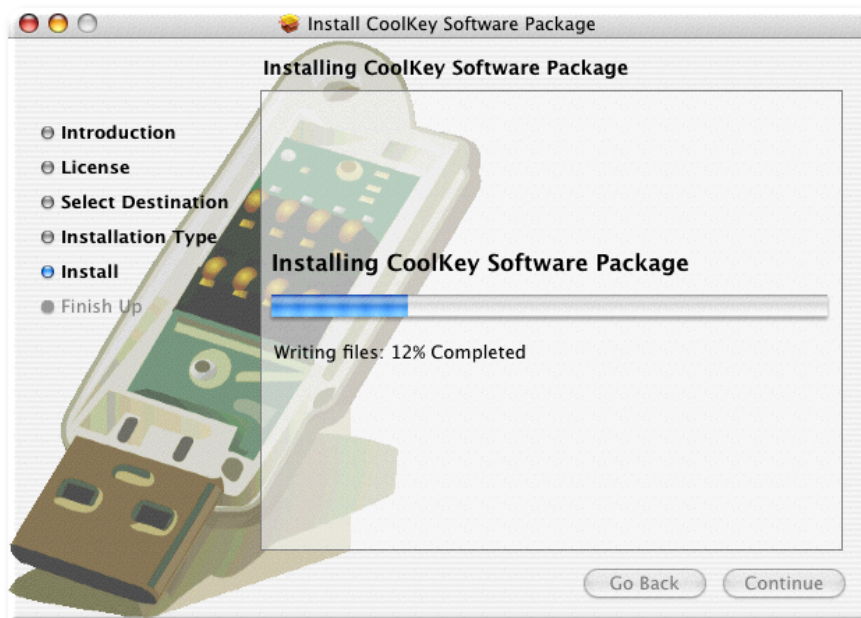


Figure 2.5. Coolkey Software Package installation in progress

When the process is complete, the e-gate token drivers, the PKCS11 module, and the TokenD software are all installed on the local system.

2.5.2. Uninstalling the Client

1. Unplug all USB tokens.
2. Stop the Enterprise Security Client.
3. Delete the **ESC.app** icon.



NOTE

There is no uninstallation program for the Mac. The **ESC.app** directory can be manually removed to remove the Enterprise Security Client.

Chapter 3. Using the Enterprise Security Client

The following sections contain basic instructions on using the Enterprise Security Client for token enrollment, formatting, and password reset operations.

3.1. Tray Icons for the Enterprise Security Client

Many programs maintain an icon in the tray or notification area which can be used to control the operation of the program, usually through context menus when the icon is right-clicked. The Enterprise Security Client provides tray icons, including tool tips for errors and actions such as inserting or removing a smart card.



Figure 3.1. Example Token Tray Icon and Tool Tip

In its default configuration, the Enterprise Security Client launches and automatically minimizes to the tray. On Red Hat Enterprise Linux, the tray icon appears only if the notification area in Gnome has been enabled.

This tray functionality behaves differently on the different operating systems:

- **Windows.** When right-clicked, the tray icon shows a simple menu with options to **Manage Smart Card**, which opens the Enterprise Security Client interface, and to **Exit Smart Card Manager**, which stops the Enterprise Security Client process. Clicking the X in the top right corner minimizes Enterprise Security Client to the tray. Double-clicking the tray icon brings Enterprise Security Client to the front. There are also notification messages, shown as standard balloon tool tips, on events like inserting or removing a card.
- **Linux.** The tray icon appears only if the notification area in Gnome has been enabled. The tray icon options are identical to the Windows options. Clicking the X in the top left corner closes the current window and minimizes Enterprise Security Client to the tray.
- **Mac.** On Mac, the tray is called the dock. Since Enterprise Security Client is based on Mozilla, right-clicking on the Enterprise Security Client dock icon reveals all the standard Mozilla Firefox menu options, including options to hide, show, and quit the client. The Enterprise Security Client also has a menu item called **Manage Smart Cards** in the dock menu, which opens the card management UI. The top level application menu has a menu under **Go, Manage Smart Card**, which also opens the card management window.

3.2. Launching Enterprise Security Client

There are two concepts for launching the Enterprise Security Client. The Enterprise Security Client process must be started and it runs silently, waiting to detect any inserted smart card or token. The user interface for the Enterprise Security Client opens automatically when smart cards are inserted or can be opened manually.

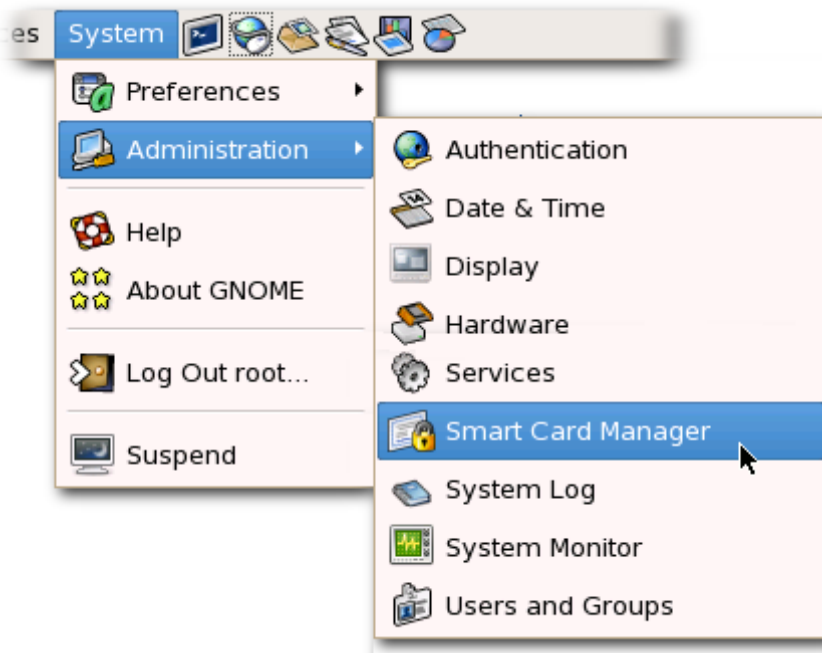
3.2.1. Opening the Enterprise Security Client on Red Hat Enterprise Linux

Initiate the Enterprise Security Client daemon (**escd**) from the command line:

```
esc
```

This daemon listens silently for smart cards and opens the GUI as soon as a smart card is inserted.

To open the Enterprise Security Client GUI manually, click **Applications**, **System Settings**, and then **Smart Card Manager**.



3.2.2. Opening the Enterprise Security Client on Microsoft Windows

Double-click the Enterprise Security Client icon on the desktop, or from the **Start** menu. The Enterprise Security Client is also configured to start on reboot.

3.2.3. Opening the Enterprise Security Client on Mac OS X

Double-click the Enterprise Security Client icon wherever the application was installed.

3.3. Configuring Phone Home

The *Phone Home* feature in the Enterprise Security Client associates information within each smart card with information that points to distinct TPS servers and Enterprise Security Client UI pages. Whenever the Enterprise Security Client accesses a new smart card, it can connect to the TPS instance and retrieve the Phone Home information.

Phone Home retrieves and then caches this information; because the information is cached locally, the TPS subsystem does not have to be contacted each time a formatted smart card is inserted.

The information can be different for every key or token, which means that different TPS servers and enrollment URLs can be configured for different corporate or customer groups. Phone Home makes it possible to configure different TPS servers for different issuers or company units, without having to configure the Enterprise Security Client manually to locate the correct server and URL.



NOTE

In order for the TPS subsystem to utilize the Phone Home feature, Phone Home must be enabled in the TPS configuration file, as follows:

```
op.format.userKey.issuerinfo.enable=true
op.format.userKey.issuerinfo.value=http://server.example.com
```

3.3.1. About Phone Home Profiles

The Enterprise Security Client is based on Mozilla XULRunner. Consequently, each user has a profile similar to the user profiles used by Mozilla Firefox and Thunderbird. The Enterprise Security Client accesses the configuration preferences file. When the Enterprise Security Client caches information for each token, the information is stored in the user's configuration file. The next time the Enterprise Security Client is launched, it retrieves the information from the configuration file instead of contacting the server again.

3.3.2. Setting Global Phone Home Information

Phone Home is triggered automatically when a security token is inserted into a machine. The system immediately attempts to read the Phone Home URL from the token and to contact the TPS server. For new tokens or for previously formatted tokens, the Phone Home information may not be available to the card.

The Enterprise Security Client configuration file, **esc-prefs.js**, has a parameter which allows a global Phone Home URL default to be set. This parameter is **esc.global.phone.home.url** and is not in the file by default.

To define the global Phone Home URL:

1. Remove any existing Enterprise Security Client user profile directory. Profile directories are created automatically when a smart card is inserted.
 - ▶ On Red Hat Enterprise Linux, the profile directory is **~/.redhat/esc**.
 - ▶ On Windows, the profile directory is **C:/Documents and Settings/user_name/Application Data/RedHat/ESC**.
 - ▶ On Mac, the profile directory is **~/Library/Application Support/ESC**.
2. Open the **esc-prefs.js** file.
 - ▶ On Red Hat Enterprise Linux 5 (32-bit), the profile directory is **/usr/lib/esc-1.1.0/defaults/preferences**. On 64-bit systems, this is **/usr/lib64/esc-1.1.0/defaults/preferences**.
 - ▶ On Windows, the profile directory is **C:/Program Files/Red Hat/ESC/defaults/preferences**.
 - ▶ On Mac, the profile directory is **/Applications/ESC.app/Contents/Resources/defaults/preferences**.
3. Add the global Phone Home parameter line to the **esc-prefs.js** file. For example:

```
pref("esc.global.phone.home.url", "http://server.example.com:7888/cgi-bin/home/index.cgi");
```

The URL can reference a machine name, a fully-qualified domain name, or an IPv4 or IPv6

address, depending on the DNS and network configuration.

When a smart card is inserted and Phone Home is launched, the Enterprise Security Client first checks the token for the Phone Home information. If no information is on the token, then the client checks the **esc-prefs.js** file for the **esc.global.phone.home.url** parameter.

If no Phone Home information is stored on the token and there is no global Phone Home parameter, the user is prompted for the Phone Home URL when a smart card is inserted, as shown in [Figure 3.2, “Prompt for Phone Home Information”](#). The other information is supplied and stored when the token is formatted. In this case, the company supplies the specific Phone Home URL for the user. After the user submits the URL, the format process adds the rest of the information to the Phone Home profile. The format process is not any different for the user.

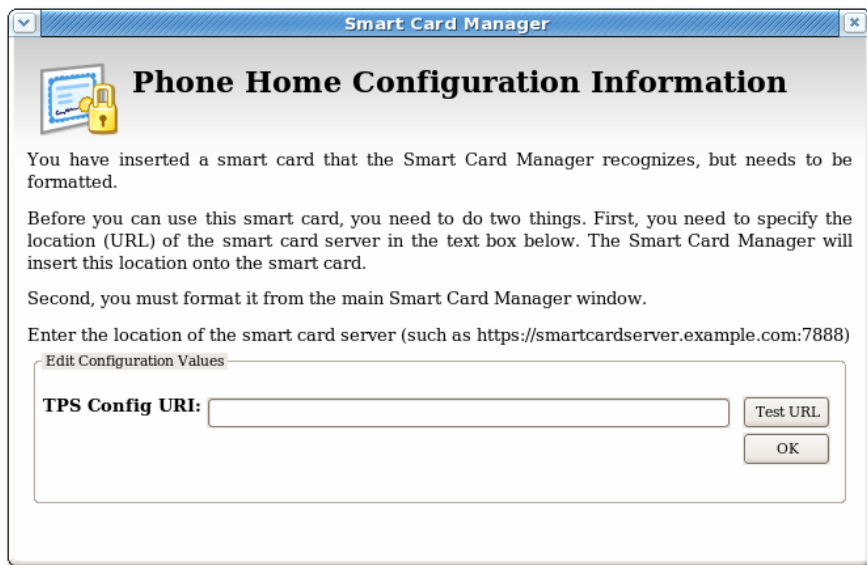


Figure 3.2. Prompt for Phone Home Information

3.3.3. Adding Phone Home Information to a Token Manually

The Phone Home information can be manually put on a token in one of two ways:

- » The preferred method is that the information is burned onto the token at the factory. When the tokens are ordered from the manufacturer, the company should also supply detailed information on how the tokens should be configured when shipped.
- » If tokens are blank, the company IT department can supply the information when formatting small groups of tokens.

The following information is used by the Phone Home feature for each smart card in the **~/.redhat/esc/alphabetic_string.default/prefs.js** file:

- » The TPS server and port. For example:

```
"esc.key.token_ID.tps.url" = "http://server.example.com:7888/nk_service"
```

- » The TPS enrollment interface URL. For example:

```
"esc.key.token_ID.tps.enrollment-ui.url" =  
"http://server.example.com:7888/cgi_bin/esc.cgi?"
```

- The issuing company name or ID. For example:

```
"esc.key.token_ID.issuer.name" = "Example Corp"
```

- The Phone Home URL. For example:

```
"esc.key.token_ID.phone.home.url" = "http://server.example.com:7888/cgi-bin/home/index.cgi?"
```

- Optionally, a default browser URL to access when an enrolled smart card is inserted.

```
"esc.key.token_ID.EnrolledTokenBrowserURL" = "http://www.test.example.com"
```

More of the parameters used by the **prefs.js** file are listed in [Table 6.2, “prefs.js Parameters”](#).



NOTE

The URLs for these parameters can reference a machine name, a fully-qualified domain name, or an IPv4 or IPv6 address, depending on the DNS and network configuration.

3.3.4. Configuring the TPS to Use Phone Home

The Phone Home feature and the different type of information used by it only work when the TPS has been properly configured to use Phone Home. If the TPS is not configured for Phone Home, then this feature is ignored. Phone Home is configured in the **index.cgi** in the **/var/lib/pki-tps/cgi-bin/home** directory; this prints the Phone Home information to XML.

[Example 3.1 “TPS Phone Home Configuration File”](#) shows an example XML file used by the TPS subsystem to configure the Phone Home feature.

Example 3.1. TPS Phone Home Configuration File

```
<ServiceInfo><IssuerName>Example Corp</IssuerName>
  <Services>
    <Operation>http://server.example.com:7888/nk_service ## TPS server URL
    </Operation>
    <UI>http://server.example.com:7888/cgi_bin/esc.cgi    ## Optional
    Enrollment UI
    </UI>
    <EnrolledTokenBrowserURL>http://www.test.url.com    ## Optional
    enrolled token url
    </EnrolledTokenBrowserURL>
  </Services>
</ServiceInfo>
```

The TPS configuration URI is the URL of the TPS server which returns the rest of the Phone Home information to the Enterprise Security Client. An example of this URL is **http://server.example.com:7888/cgi-bin/home/index.cgi**; the URL can reference the machine name, fully-qualified domain name, or an IPv4 or IPv6 address, as appropriate. When the TPS configuration URI is accessed, the TPS server is prompted to return all of the Phone Home information to the Enterprise Security Client.

To test the URL of the Smart Card server, enter the address in the **TPS Config URI** field, and click **Test URL**.

If the server is successfully contacted, a message box indicates success. If the test connection fails, an error dialog appears.

3.4. Setting up Users to Be Enrolled

When the Token Processing System is installed, one of its configuration settings is the LDAP directory which contains the users who are allowed to enroll a token. Only users who are stored within this authentication directory are allowed to enroll, format, or have a token. Before attempting to enroll a token or smart card, make sure that the person requesting the operation has an entry in the LDAP directory.

The TPS is configured to look at a specific base DN in the LDAP directory. This is configured in the TPS's **CS.cfg**:

```
auth.instance.0.baseDN=dc=example,dc=com
auth.instance.0.hostport=server.example.com:389
```

For a user to be allowed to enroll a token, the user must be somewhere below the base DN.

If the user does not already have an entry, then the administrator must add the user to the specified LDAP directory in the specified base DN before any tokens can be enrolled for the user.

```
/usr/bin/ldapmodify -a -D "cn=Directory Manager" -w secret -p 389 -h
server.example.com

dn: uid=jsmith,ou=People, dc=example,dc=com
objectclass: person
objectclass: inetorgperson
objectclass: top
uid: jsmith
cn: John Smith
email: jsmith@example.com
userPassword: secret
```

3.5. Enrolling a Smart Card Automatically

Because the Enterprise Security Client is configured using the Phone Home feature, enrolling a smart card is extremely easy. Because the information needed to contact the backend TPS server is provided with each smart card, the user is guided quickly and easily through the procedure.

To enroll an uninitialized smart card:



NOTE

This procedure assumes that the smart card is uninitialized and the appropriate Phone Home information has been configured.

1. Ensure that the Enterprise Security Client is running.
2. Insert an uninitialized smart card, pre-formatted with the Phone Home information for the TPS and the enrollment interface URL for the user's organization.

The smart card can be added either by placing a USB form factor smart card into a free USB slot, or by inserting a standard, full-sized smart card into a smart card reader.

When the system recognizes the smart card, it displays a message indicating it has detected an uninitialized smart card.



3. Click **Enroll My Smart Card Now** to display the smart card enrollment form.

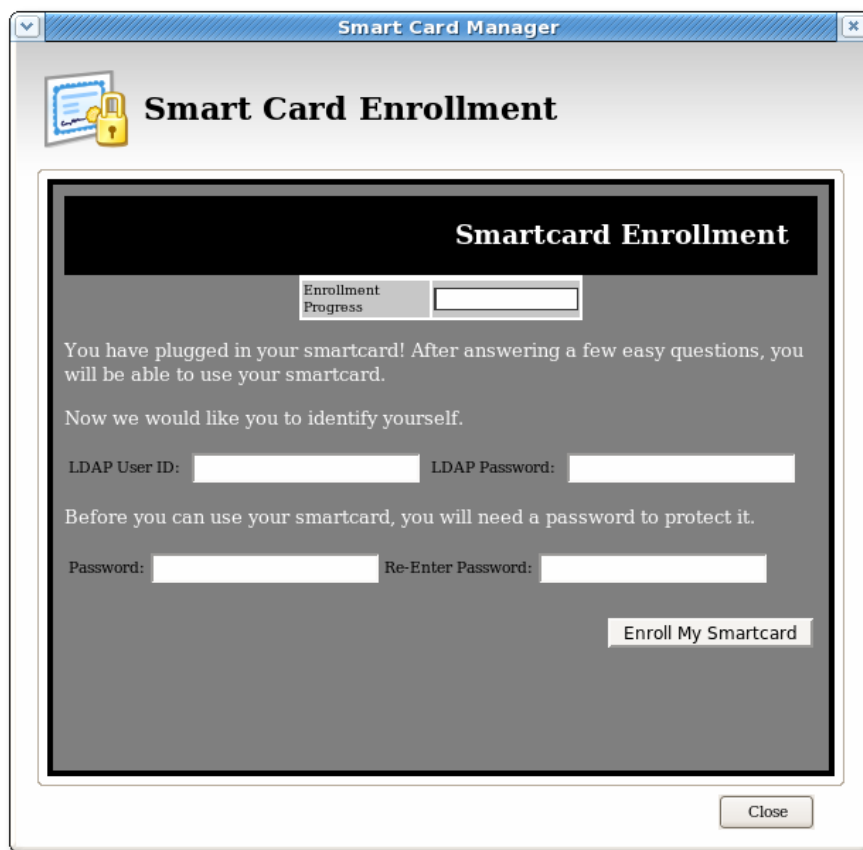


NOTE

If you remove the card at this point, a message displays stating that the smart card can no longer be detected. Reinsert the card to continue with the enrollment process.

The enrollment files are accessed remotely; they reside on the TPS instance. If the network connection is bad or broken, then, an error may come up saying *Check the Network Connection and Try Again*. It is also possible that the enrollment window appears to open but the enrollment process doesn't proceed. The enrollment pages can be cached if the Enterprise Security Client previously connect to them successfully, so the enrollment UI opens even if the network is offline. Try restarting Enterprise Security Client and check the network connection.

4. Because the Smart Card Manager now knows where the enrollment UI is located (it is included in the Phone Home information), the enrollment form is displayed for the user to enter the required information.



The image shows a screenshot of the 'Smart Card Manager' application window. The title bar reads 'Smart Card Manager'. Inside the window, there is a sub-header 'Smart Card Enrollment' with a small icon of a smart card and a lock. Below this, there is a progress indicator labeled 'Enrollment Progress' with a small progress bar. The main text area contains the following instructions: 'You have plugged in your smartcard! After answering a few easy questions, you will be able to use your smartcard.' and 'Now we would like you to identify yourself.' There are two input fields: 'LDAP User ID:' and 'LDAP Password:'. Below these, another instruction reads: 'Before you can use your smartcard, you will need a password to protect it.' There are two more input fields: 'Password:' and 'Re-Enter Password:'. At the bottom right of the main area is a button labeled 'Enroll My Smartcard'. At the bottom right of the entire window is a 'Close' button.

This illustration shows the default enrollment UI included with the TPS server. This UI is a standard HTML form, which you can customize to suit your own deployment requirements. This could include adding a company logo or adding and changing field text.

See [Section 6.4, “Customizing the Smart Card Enrollment User Interface”](#) for information on customizing the UI.

5. The sample enrollment UI requires the following information for the TPS server to process the smart card enrollment operation:
 - » *LDAP User ID*. This is the LDAP user ID of the user enrolling the smart card; this can also be a screen name or employee or customer ID number.
 - » *LDAP Password*. This is the password corresponding to the user ID entered; this can be a simple password or a customer number.



NOTE

The LDAP user ID and password are related to the Directory Server user. The TPS server is usually associated with a Directory Server, which stores user information and through which the TPS authenticates users. Passwords must conform to the password policy configured in the Directory Server.



NOTE

If the password is stored using the SSHA hash, then any exclamation point (!) and dollar sign (\$) characters in the password must be properly escaped for a user to bind successfully to the Enterprise Security Client on Windows XP and Vista systems.

- For the dollar sign (\$) character, escape the dollar sign *when the password is created*:

\\$

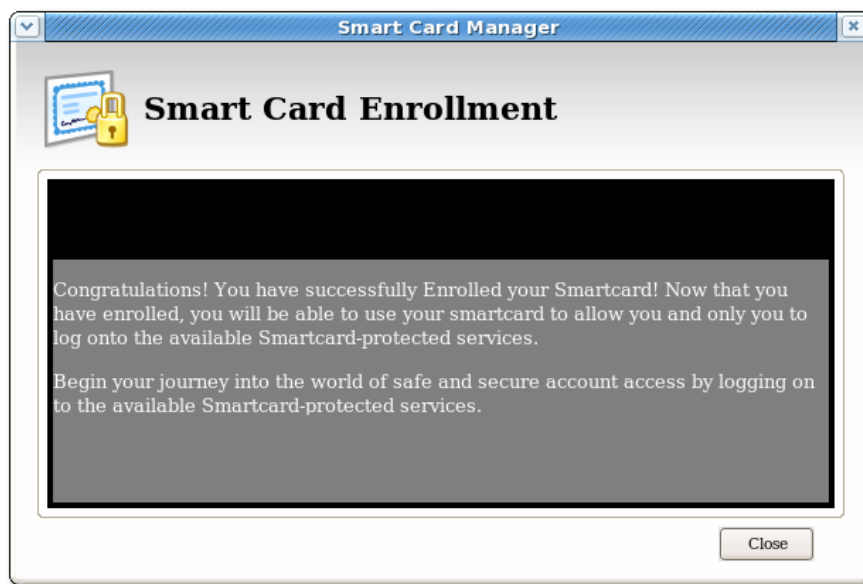
Then, enter only the dollar sign (\$) character when logging into the Enterprise Security Client.

- For the exclamation point (!) character, escape the character when the password is created *and* when the password is entered to log into the Enterprise Security Client.

\!

- **Password and Re-Enter Password.** These fields set and confirm the smart card's password, used to protect the card information.

6. After you have entered all required information, click **Enroll My Smart Card** to submit the information and enroll the card.
7. When the enrollment process is complete, a message page opens which shows that the card was successfully enrolled and can offer custom instructions on using the newly-enrolled smart card.



3.6. Managing Smart Cards

You can use the **Manage Smart Cards** page to perform many of the operations that can be applied to one of the cryptographic keys stored on the token.

You can use this page to format the token, set and reset the card's password, and to display card information. Two other operations, enrolling tokens and viewing the diagnostic logs, are also accessed through the **Manage Smart Cards** page. These operations are addressed in other sections.

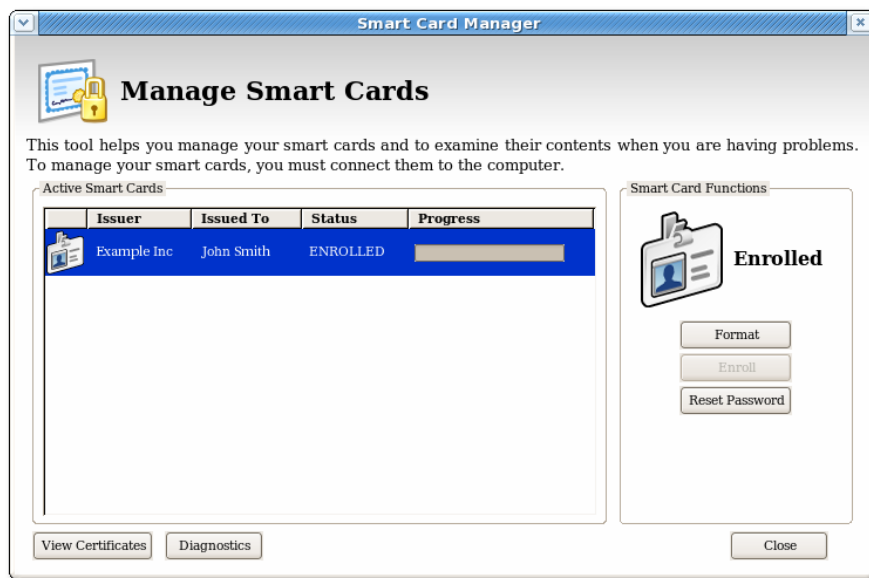


Figure 3.3. Manage Smart Cards Page

3.6.1. Formatting the Smart Card

When you format a smart card, it is reset to the uninitialized state. This removes all previously generated user key pairs and erases the password set on the smart card during enrollment.

The TPS server can be configured to load newer versions of the applet and symmetric keys onto the card. The TPS supports the CoolKey applet which is shipped with Red Hat Enterprise Linux 5.3.

NOTE

An e-gate token cannot be formatted on a Windows machine using the Enterprise Security Client. To format an e-gate card, perform the operation on Red Hat Enterprise Linux or Mac.

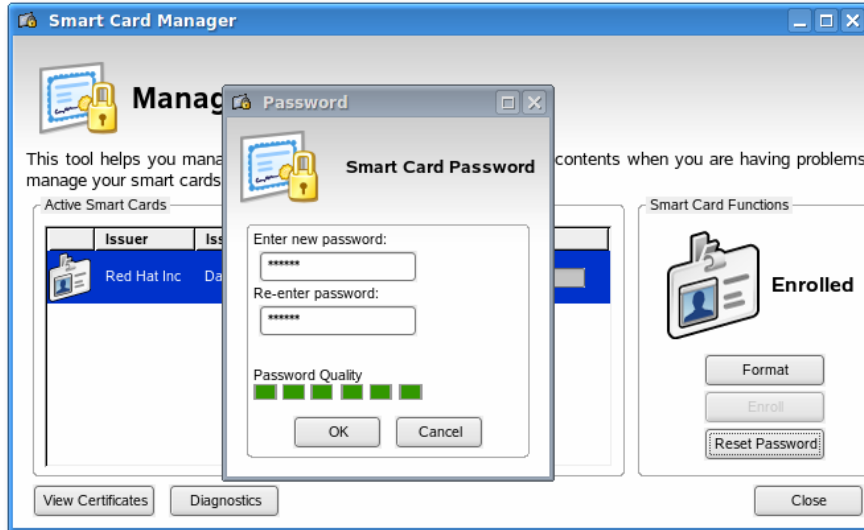
To format a smart card:

1. Insert a supported smart card into the computer. Ensure that the card is listed in the **Active Smart Cards** table.
2. In the **Smart Card Functions** section of the **Manage Smart Cards** screen, click **Format**.
3. If the TPS has been configured for user authentication, enter the user credentials in the authentication dialog, and click **Submit**.
4. During the formatting process, the status of the card changes to BUSY and a progress bar is displayed. A success message is displayed when the formatting process is complete. Click **OK** to close the message box.
5. When the formatting process is complete, the **Active Smart Cards** table shows the card status as UNINITIALIZED.

3.6.2. Resetting a Smart Card Password

If a user forgets the password for a smart card after the card is enrolled, it is possible to reset the password. To reset the password on a smart card:

1. Insert a supported smart card into the computer. Ensure that the card is listed in the **Active Smart Cards** table.
2. In the **Smart Card Functions** section of the Manage Smart Cards screen, click **Reset Password** to display the **Password** dialog.
3. Enter a new smart card password in the **Enter new password** field.
4. Confirm the new smart card password in the **Re-Enter password** field, and then click **OK**.



5. If the TPS has been configured for user authentication, enter the user credentials in the authentication dialog, and click **Submit**.
6. Wait for the password to finish being reset.

3.6.3. Viewing Certificates

The **Smart Card Manager** can display basic information about a selected smart card, including stored keys and certificates. To view certificate information:

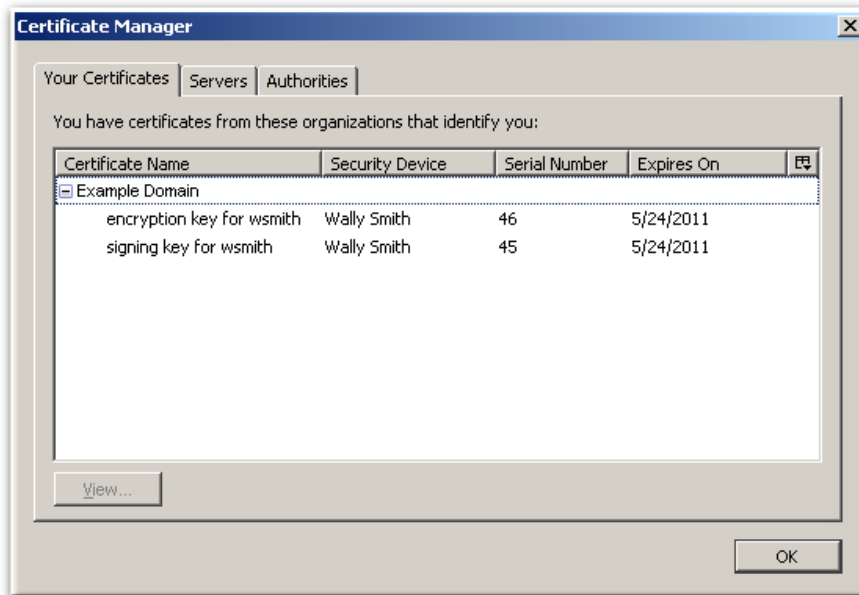
1. Insert a supported smart card into the computer. Ensure that the card is listed in the **Active Smart Cards** table.
2. Select the card from the list, and click **View Certificates**.



This displays basic information about the certificates stored on the card, including the serial

number, certificate nickname, and validity dates.

3. To view more detailed information about a certificate, select the certificate from the list and click **View**.



3.6.4. Importing CA Certificates

The Xulrunner Gecko engine implements stringent controls over which SSL-based URLs can be visited by client like a browser or the Enterprise Security Client. If the Enterprise Security Client (through the Xulrunner framework) does not trust a URL, the URL can not be visited.

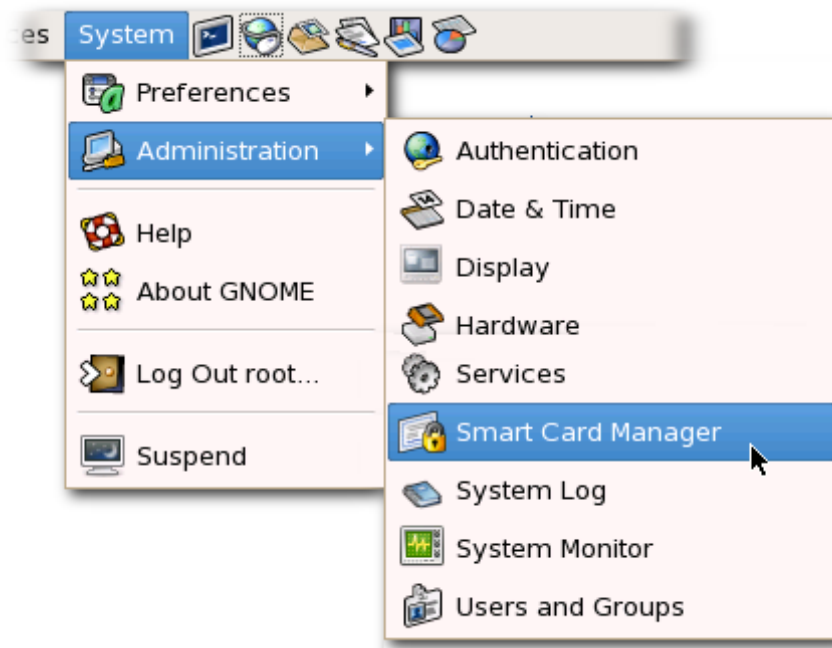
One way to trust an SSL-based URL is to import and trust the CA certificate chain of the CA which issued the certificates for the site. (The other is to create a trust *security exception* for the site, as in [Section 3.6.5, “Adding Exceptions for Servers”](#).)

Any CA which issues certificates for smart cards must be trusted by the Enterprise Security Client application, which means that its CA certificate must be imported into the Enterprise Security Client.

1. Open the CA's end user pages in a web browser.

`https://server.example.com:9444/ca/ee/ca/`

2. Click the **Retrieval** tab at the top.
3. In the left menu, click the **Import CA Certificate Chain** link.
4. Choose the radio button to download the chain as a file, and remember the location and name of the downloaded file.
5. Open the Enterprise Security Client.

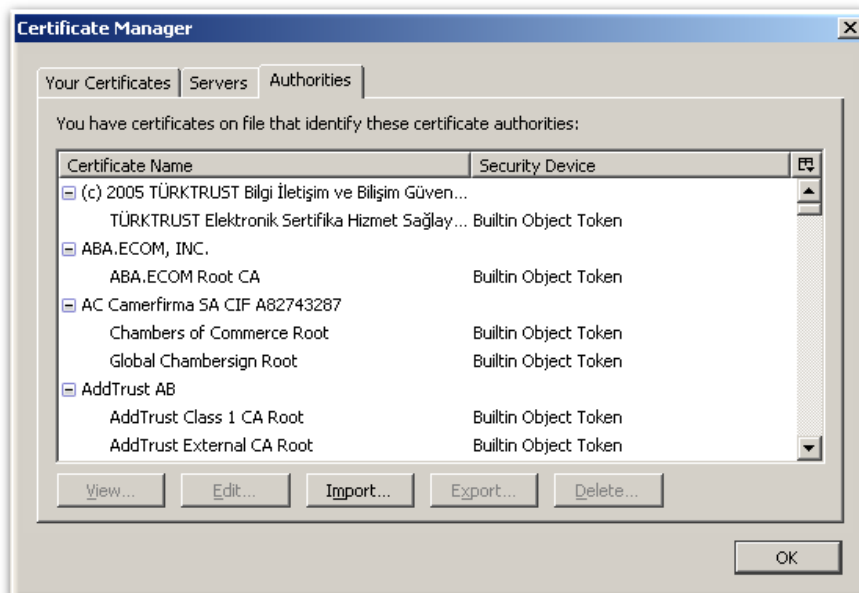


6. Click the **View Certificates** button.



7. Click the **Authorities** tab.

8. Click **Import**.



9. Browse to the CA certificate chain file, and select it.
10. When prompted, confirm that you want to trust the CA.

3.6.5. Adding Exceptions for Servers

The Xulrunner Gecko engine implements stringent controls over which SSL-based URLs can be visited by client like a browser or the Enterprise Security Client. If the Enterprise Security Client (through the Xulrunner framework) does not trust a URL, the URL can not be visited.

One way to trust an SSL-based URL is to create a trust *security exception* for the site, which imports the certificate for the site and forces the Enterprise Security Client to recognize it. (The other option is to import the CA certificate chain for the site and automatically trust it, as in [Section 3.6.4, “Importing CA Certificates”](#).)

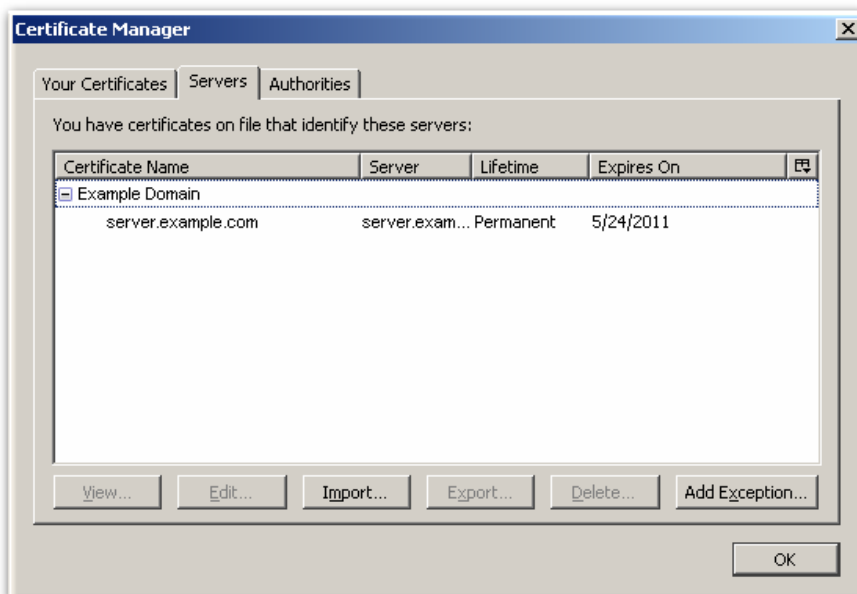
The smart card may be used to access services or websites over SSL that require special security exceptions; these exceptions can be configured through the Enterprise Security Client, similar to configuring exceptions for websites in a browser like Mozilla Firefox.

1. Open the Enterprise Security Client.

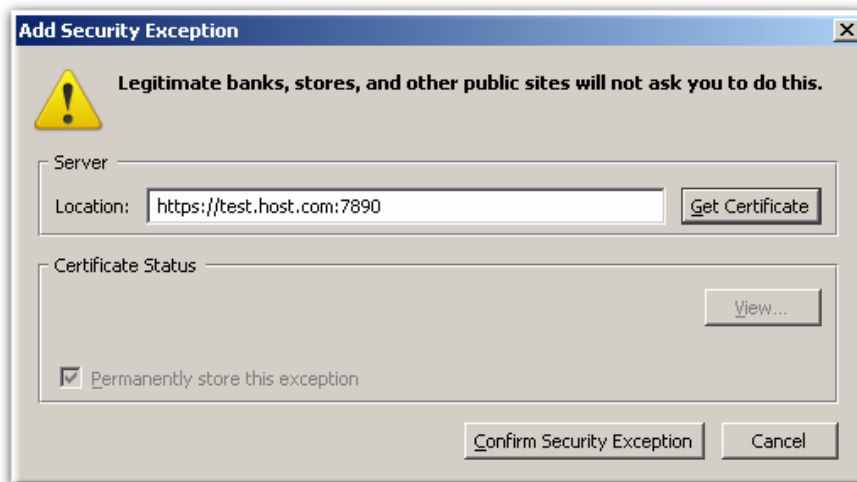
2. Click the **View Certificates** button.



3. Click the **Servers** tab.
4. Click **Add Exception**.



5. Enter the URL, including any port numbers, for the site or service which the smart card will be used to access. Then click the **Get Certificates** button to download the server certificate for the site.



6. Click **Confirm Security Exception** to add the site to the list of allowed sites.

3.6.6. Enrolling Smart Cards

Most smart cards will be automatically enrolled using the automated enrollment procedure, described in [Section 3.5, “Enrolling a Smart Card Automatically”](#). You can also use the **Manage Smart Cards** facility to manually enroll a smart card.

If you enroll a token with the user key pairs, then the token can be used for certificate-based operations such as SSL client authentication and S/MIME.



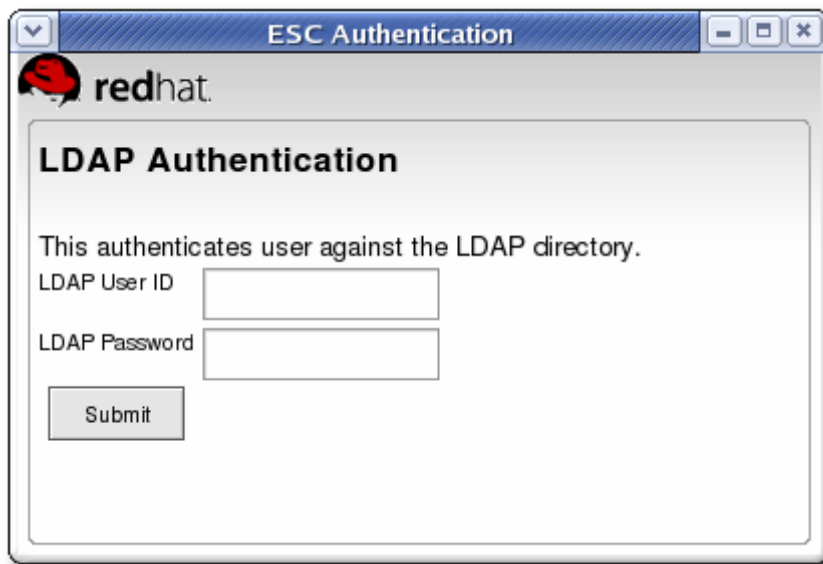
NOTE

The TPS server can be configured to generate the user key pairs on the server and then archived in the DRM subsystem for recovery if the token is lost.

To enroll a smart card manually:

1. Insert a supported, unenrolled smart card into the computer. Ensure that the card is listed in the **Active Smart Cards** table.
2. Click **Enroll** to display the **Password** dialog.
3. Enter a new key password in the **Enter a password** field.
Confirm the new password in the **Re-Enter a password** field.
4. Click **OK** to begin the enrollment.
5. If the TPS has been configured for user authentication, enter the user credentials in the authentication dialog, and click **Submit**.

If the TPS has been configured to archive keys to the DRM, the enrollment process will begin generating and archiving keys.



When the enrollment is complete, the status of the smart card is displayed as ENROLLED.

3.6.7. Re-Enrolling Tokens

Commonly, a smart card will need to be re-enrolled while the certificates on it are still active. The smart card may be reused for a new user, or the certificates may be nearing their expiration date so the original user is having the smart card re-enrolled during the grace period.

The enrollment process does not automatically revoke active certificates, so enrolling a token without removing its active certificates will leave the old certificates active.

Before re-enrolling a token, the agent or officer must either manually revoke the certificates on the token or format the card again to return it to an uninitialized state. Formatting a token automatically removes and revokes its old certificates.

Formatting a token is described in [Section 3.6.1, "Formatting the Smart Card"](#).

3.7. Verifying That the Mac TokenD Is Working Properly

The TokenD software installed on a Macintosh computer provides a link between the Certificate System CoolKeys and the Mac CDSA security API, which provides a wide variety of security functionality.

For example, the Apple Mail application can use a *KeyChain* to perform security-related tasks. A KeyChain can store entities such as certificates, passwords, and private and public keys. Although most KeyChains are stored in software, the CDSA API allows KeyChains to be stored on smart cards or keys. CoolKey TokenD allows a Certificate System key to be displayed as a KeyChain.

To verify that the TokenD software is working correctly:

1. Ensure that the Enterprise Security Client has been installed on the Mac computer.
2. Use the Enterprise Security Client to enroll a token, enabling it with the correct certificates and key information.
3. Insert the enrolled token into a USB slot.
4. If TokenD is working, the token blinks for a few seconds while the information is obtained from the token. This is because the Mac CDSA layer is making a request for data.
5. Open the Mac **Keychain Access** utility in **Applications/Utilities/**

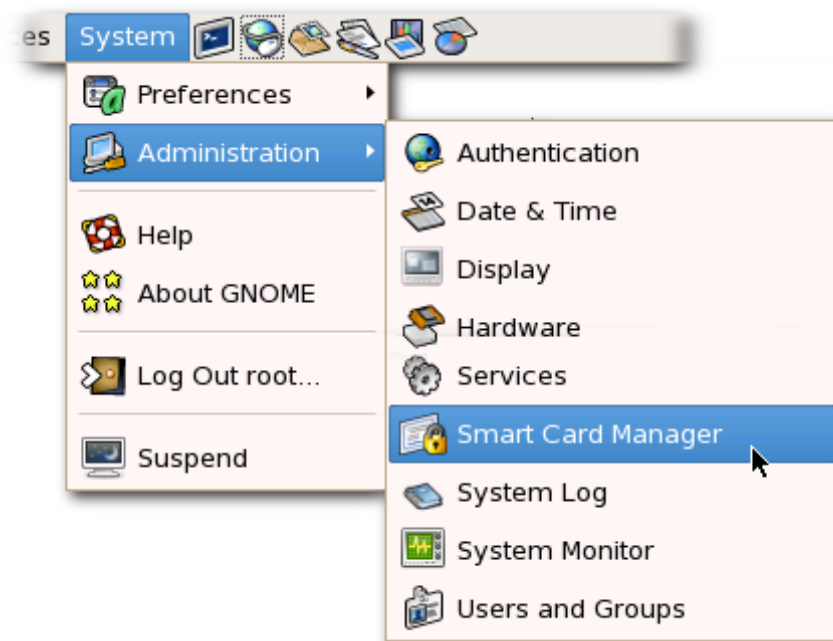
6. Find the new **\$Keychain** entry in the list of valid chains. The chain has the key's UID in its name.
7. Click the CoolKey KeyChain to view the certificates and keys on the token.

3.8. Diagnosing Problems

The Enterprise Security Client includes basic diagnostic tools and a simple interface to log errors and common events, such as inserting and removing a smart card or changing the card's password. The diagnostic tools can identify and notify users about problems with the Enterprise Security Client, smart cards, and TPS connections.

To open the **Diagnostics Information** window:

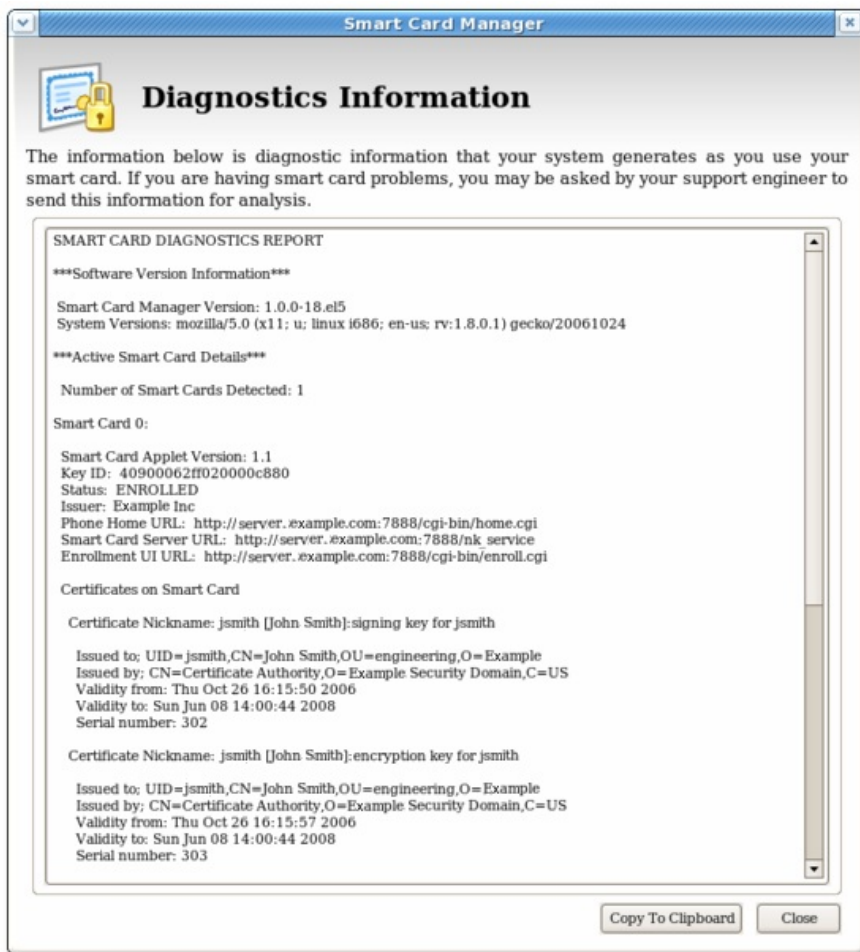
1. Open the Enterprise Security Client.



2. Select the smart card to check from the list.
3. Click the **Diagnostics** button.



4. This opens the **Diagnostic Information** window for the selected smart card.



The **Diagnostics Information** screen displays the following information:

- » The Enterprise Security Client version number.
- » The version information for the Xulrunner framework upon which the client is running.
- » The number of cards detected by the Enterprise Security Client.

For each card detected, the following information is displayed:

- » The version of the applet running on the smart card.
- » The alpha-numeric ID of the smart card.
- » The card's status, which can be any of the three things:
 - **NO_APPLET** No key was detected.
 - **UNINITIALIZED**. The key was detected, but no certificates have been enrolled.
 - **ENROLLED**. The detected card has been enrolled with certificate and card information.
- » The card's Phone Home URL. This is the URL from which all Phone Home information is obtained.
- » The card issuer name, such as **Example Corp.**
- » The card's answer-to-reset (ATR) string. This is a unique value that can be used to identify different classes of smart cards. For example:

3BEC00FF8131FE45A0000000563333304A330600A1

- » The TPS Phone Home URL.
- » The TPS server URL. This is retrieved through Phone Home.

- » The TPS enrollment form URL. This is retrieved through Phone Home.
- » Detailed information about each certificate contained on the card.
- » A running log of the most recent Enterprise Security Client errors and common events.

The Enterprise Security Client records two types of diagnostic information. It records *errors* that are returned by the smart card, and it records *events* that have occurred through the Enterprise Security Client. It also returns basic information about the smart card configuration.

3.8.1. Errors

- » The Enterprise Security Client does not recognize a card.
- » Problems occur during a smart card operation, such as a certificate enrollment, password reset, or format operation.
- » The Enterprise Security Client loses the connection to the smart card. This can happen when problems occur communicating with the **PCSC** daemon.
- » The connection between the Enterprise Security Client and TPS is lost.

Smart cards can report certain error codes to the TPS; these are recorded in the TPS's **tps-debug.log** or **tps-error.log** files, depending on the cause for the message.

Table 3.1. Smart Card Error Codes

Return Code	Description
General Error Codes	
6400	No specific diagnosis
6700	Wrong length in Lc
6982	Security status not satisfied
6985	Conditions of use not satisfied
6a86	Incorrect P1 P2
6d00	Invalid instruction
6e00	Invalid class
Install Load Errors	
6581	Memory Failure
6a80	Incorrect parameters in data field
6a84	Not enough memory space
6a88	Referenced data not found
Delete Errors	
6200	Application has been logically deleted
6581	Memory failure
6985	Referenced data cannot be deleted
6a88	Referenced data not found
6a82	Application not found
6a80	Incorrect values in command data
Get Data Errors	
6a88	Referenced data not found
Get Status Errors	
6310	More data available
6a88	Referenced data not found
6a80	Incorrect values in command data
Load Errors	
6581	Memory failure
6a84	Not enough memory space
6a86	Incorrect P1/P2
6985	Conditions of use not satisfied

3.8.2. Events

- Simple events such as card insertions and removals, successfully completed operations, card operations that result in an error, and similar events.
- Errors are reported from the TPS to the Enterprise Security Client.
- The NSS crypto library is initialized.
- Other low-level smart card events are detected.

Chapter 4. Using Security Officer Mode

The Enterprise Security Client, together with the TPS subsystem, supports a special *security officer* mode of operation. This mode allows a supervisory individual, a security officer, the ability to oversee the face to face enrollment of regular users in a given organization.

Security officer mode provides the ability to enroll individuals under the supervision of a security officer, a designated user-type who can manage other user's smart cards in face-to-face and very secure operations. Security officer mode overlaps with some regular user operations, with additional security features:

- ▶ The ability to search for an individual within an organization.
- ▶ An interface that displays a photo and other pertinent information about an individual.
- ▶ The ability to enroll approved individuals.
- ▶ Formatting or resetting a user's card.
- ▶ Formatting or resetting a security officer's card.
- ▶ Enrolling a temporary card for a user that has misplaced their primary card.
- ▶ Storing TPS server information on a card. This Phone Home information is used by the Enterprise Security Client to contact a given TPS server installation.

Working in the security officer mode falls into two distinct areas:

- ▶ Creating and managing security officers.
- ▶ Managing regular users by security officers.

When security officer mode is enabled, the Enterprise Security Client uses an external user interface provided by the server. This interface takes control of smart card operations in place of the local XUL code that the Enterprise Security Client normally uses.

The external interface maintains control until security officer mode is disabled.



TIP

It is a good idea to run security officer clients over SSL, so make sure that the TPS is configured to run in SSL, and then point the Enterprise Security Client to the TPS's SSL agent port.

4.1. Enabling Security Officer Mode

There are two areas where the security officer mode must be configured, both in the TPS and in the Enterprise Security Client's **esc-prefs.js** file.

In the TPS:

1. Add the security officer user entry to the TPS database as a member of the TUS Officers group.



TIP

It can be simpler to add and copy user entries in the LDAP database using the Red Hat Directory Server Console. Using the Directory Server Console is described more in the *Red Hat Directory Server Administrators Guide* in [section 3.1.2, "Creating Directory Entries."](#)

There are two subtrees associated with the TPS. One is for external users, which has a distinguished name (DN) like **dc=server,dc=example,dc=com**; this directory is used to authenticate any user attempting to enroll a smart card. The other database is used for internal TPS instance entries, including TPS agents, administrators, and security officers. This subtree has a DN like **dc=server.example.com-pki-tps**. The TUS Officers group entry is under the **dc=server.example.com-pki-tps** suffix.

Any security officer entry has to be a child entry of the TUS Officers group entry. This means that the group entry is the main entry, and the user entry is directly beneath it in the directory tree.

The TUS Officers group entry is **cn=TUS**

Officers,ou=Groups,dc=server.example.com-pki-tps.

For example, to add the security officer entry using **ldapmodify**:

```
/usr/lib/mozldap/ldapmodify -a -D "cn=Directory Manager" -w secret -p 389 -h
server.example.com

dn: uid=jsmith,cn=TUS Officers,ou=Groups, dc=server.example.com-pki-tps
objectclass: inetorgperson
objectclass: organizationalPerson
objectclass: person
objectclass: top
sn: smith
uid: jsmith
cn: John Smith
mail: jsmith@example.com
userPassword: secret
```

Press the **Enter** key twice to send the entry, or use **Ctrl+D**.

2. Check the TPS's security officer workstation to make sure it is pointing to the external user subtree DN to authenticate users who will enroll smart cards.

```
vim /var/lib/pki-tps/cgi-bin/sow/cfg.pl

#
# Feel free to modify the following parameters:
#
my $ldapHost = "localhost";
my $ldapPort = "389";
my $basedn = "ou=People,dc=server, dc=example,dc=com";
my $port = "7888";
my $secure_port = "7889";
my $host = "localhost";
```

Then, configure the Enterprise Security Client.

1. First, trust the CA certificate chain.
 - a. Open the CA's end-entities page.

<https://server.example.com:9444/ca/ee/ca/>

- b. Click the **Retrieval** tab, and download the CA certificate chain.
 - c. Open the Enterprise Security Client.

esc

- d. Click the **View Certificates** button.
 - e. Click the **Authorities** tab.
 - f. Click the **Import** button, and import the CA certificate chain.
 - g. Set the trust settings for the CA certificate chain.
2. Then, format and enroll the security officer's token. This token is used to access the security officer Enterprise Security Client UI.
 - a. Insert a blank token.
 - b. When the prompt for the Phone Home information opens, enter the security officer URL.

```
/var/lib/pki-tps/cgi-bin/so/index.cgi
```

- c. Click the **Format** button to format the security officer's token.
- d. Close the interface and stop the Enterprise Security Client.
- e. Add two parameters in the **esc-prefs.js** file. The first, **esc.disable.password.prompt**, sets security officer mode. The second, **esc.security.url**, points to the security officer enrollment page. Just the presence of the **esc.security.url** parameter instructs the Enterprise Security Client to open in security officer mode next time it opens.

```
pref("esc.disable.password.prompt", "no");
pref("esc.security.url", "https://server.example.com:7888/cgi-
bin/so/enroll.cgi");
```

- f. Start the Enterprise Security Client again, and open the UI.

```
esc
```

- g. The Enterprise Security Client is configured to connect to the security officer enrollment form in order to enroll the new security officer's token. Enroll the token as described in [Section 4.2, "Enrolling a New Security Officer"](#).
- h. Close the interface and stop the Enterprise Security Client.
- i. Edit the **esc-prefs.js** file again, and this time change the **esc.security.url** parameter to point to the security officer workstation page.

```
pref("esc.security.url", "https://server.example.com:7889/cgi-
bin/sow/welcome.cgi");
```

- j. Restart the Enterprise Security Client again. The UI now points to the security officer workstation to allow security officers to enroll tokens for regular users.

To disable security officer mode, close the Enterprise Security Client GUI, stop the **escd** process, and comment out the **esc.security.url** and **esc.disable.password.prompt** lines in the **esc-prefs.js** file. When the **esc** process is restarted, it starts in normal mode.

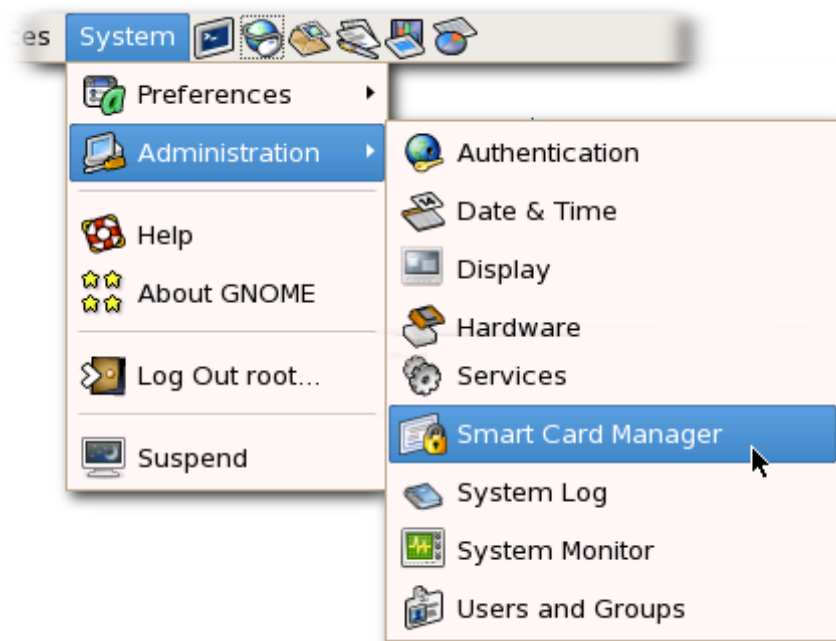
4.2. Enrolling a New Security Officer

Security officers are set up using a separate, unique interface than the one for regular enrollments or the one used for security officer-managed enrollments.

1. Make sure the **esc** process is running.

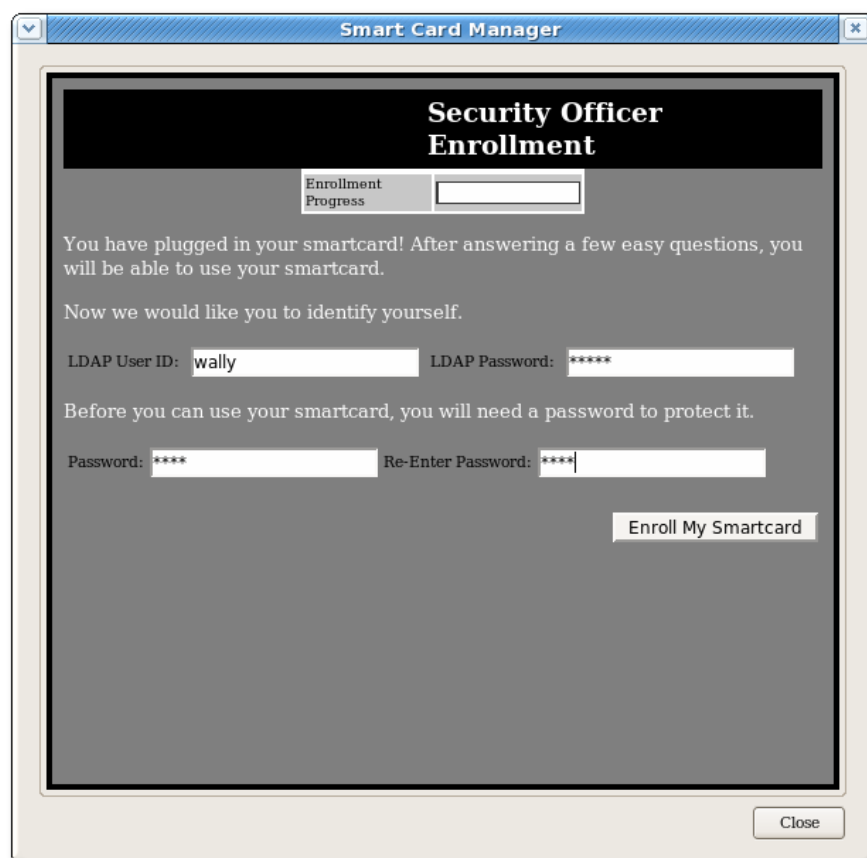
esc

Then open the Enterprise Security Client UI.



With security officer mode enabled in the `esc-pref.js` file ([Section 4.1, “Enabling Security Officer Mode”](#)) the security officer enrollment page opens.

2. In the **Security Officer Enrollment** window, enter the LDAP user name and password of the new security officer and a password that will be used with the security officer's smart card.



**NOTE**

If the password is stored using the SSHA hash, then any exclamation point (!) and dollar sign (\$) characters in the password must be properly escaped for a user to bind successfully to the Enterprise Security Client on Windows XP and Vista systems.

- For the dollar sign (\$) character, escape the dollar sign *when the password is created*:

\\$

Then, enter only the dollar sign (\$) character when logging into the Enterprise Security Client.

- For the exclamation point (!) character, escape the character when the password is created *and* when the password is entered to log into the Enterprise Security Client.

\!

3. Click **Enroll My Smartcard**.

This produces a smart card which contains the certificates needed by the security officer to access the Enterprise Security Client security officer, so that regular users can be enrolled and managed within the system.

4.3. Using Security Officers to Manage Users

The security officer Station page manages regular users through operations such as enrolling new or temporary cards, formatting cards, and setting the Phone Home URL.

4.3.1. Enrolling a New User

There is one significant difference between enrolling a user's smart card in security officer mode and the process in [Section 3.5, “Enrolling a Smart Card Automatically”](#) and [Section 3.6.6, “Enrolling Smart Cards”](#). All processes require logging into an LDAP database to verify the user's identity, but the security officer mode has an extra step to compare some credentials presented by the user against some information in the database (such as a photograph).

1. Make sure the **esc** process is running. If necessary, start the process.

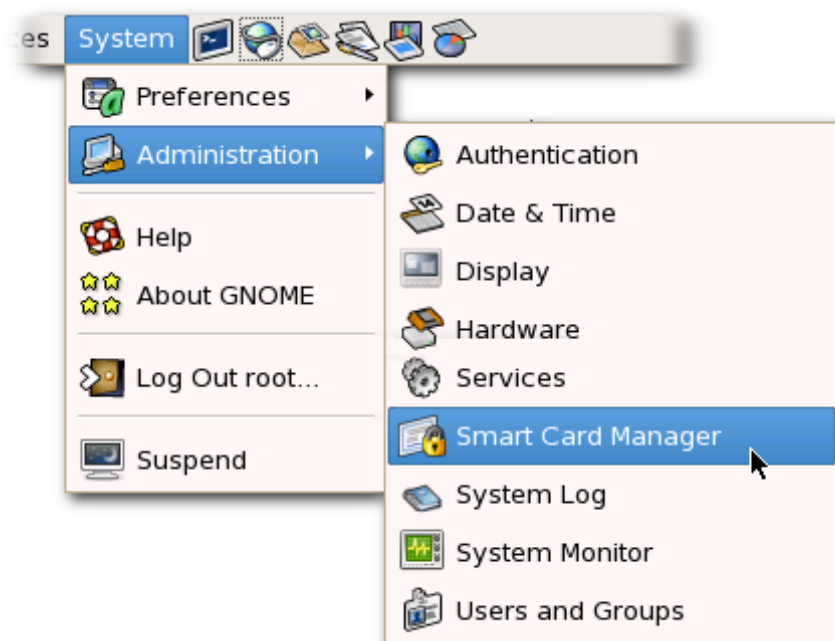
esc

Also, make sure that security officer mode is enabled, as described in [Section 4.1, “Enabling Security Officer Mode”](#).

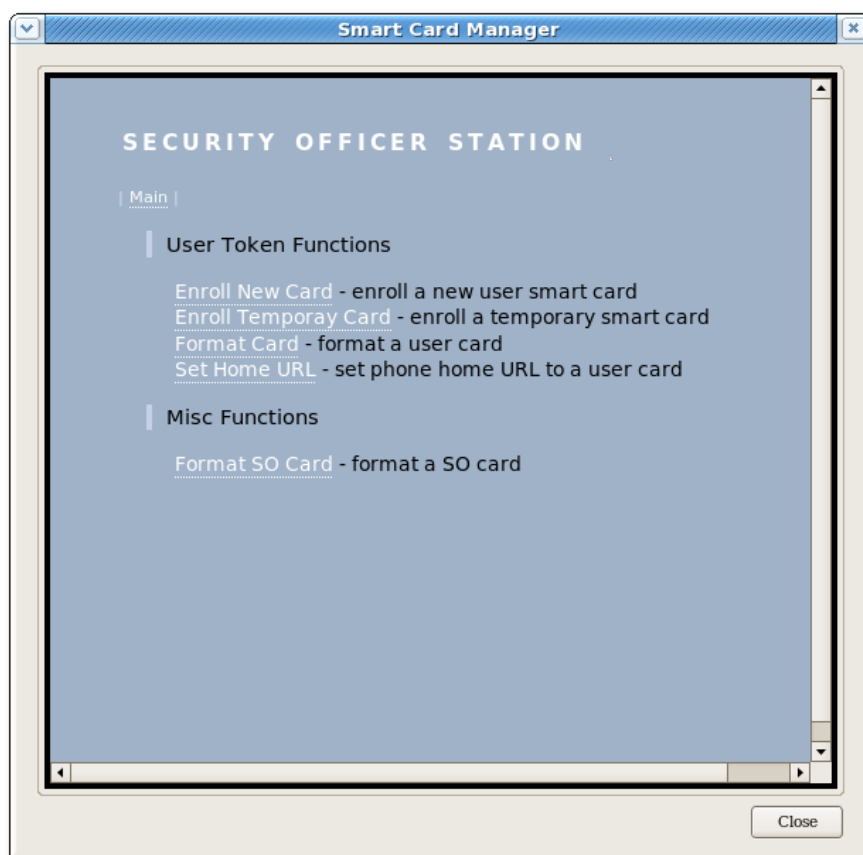
2. Then open the Enterprise Security Client UI.

**NOTE**

Ensure that there is a valid and enrolled security officer card plugged into the computer. A security officer's credentials are required to access the following pages.



3. Click **Continue** to display the security officer Station page. The client may prompt for the password for the security officer's card (which is required for SSL client authentication) or to select the security officer's signing certificate from the drop-down menu.
4. Click the **Enroll New Card** link to display the **Security Officer Select User** page.



5. Enter the LDAP name of the user who is to receive a new smart card.
6. Click **Continue**. If the user exists, the **Security Officer Confirm User** page opens.
7. Compare the information returned in the Enterprise Security Client UI to the person or credentials that are present.
8. If all the details are correct, click **Continue** to display the **Security Officer Enroll User**

page. This page prompts the officer to insert a new smart card into the computer.

9. If the smart card is properly recognized, enter the new password for this card and click **Start Enrollment**.

A successful enrollment produces a smart card that a user can use to access the secured network and services for which the smart card was made.

4.3.2. Performing Other Security Officer Tasks

All of the other operations that can be performed for regular users by a security officer — issuing temporary tokens, re-enrolling tokens, or setting a Phone Home URL — are performed as described in [Chapter 3, Using the Enterprise Security Client](#), after opening the security officer UI.

1. Make sure the **esc** process is running. If necessary, start the process.

```
esc
```

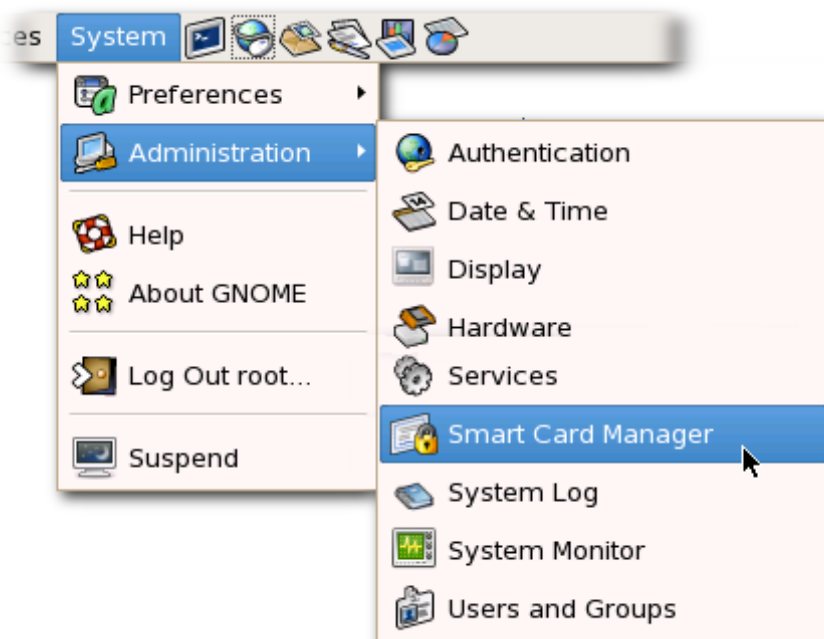
Also, make sure that security officer mode is enabled, as described in [Section 4.1, “Enabling Security Officer Mode”](#).

2. Then open the Enterprise Security Client UI.

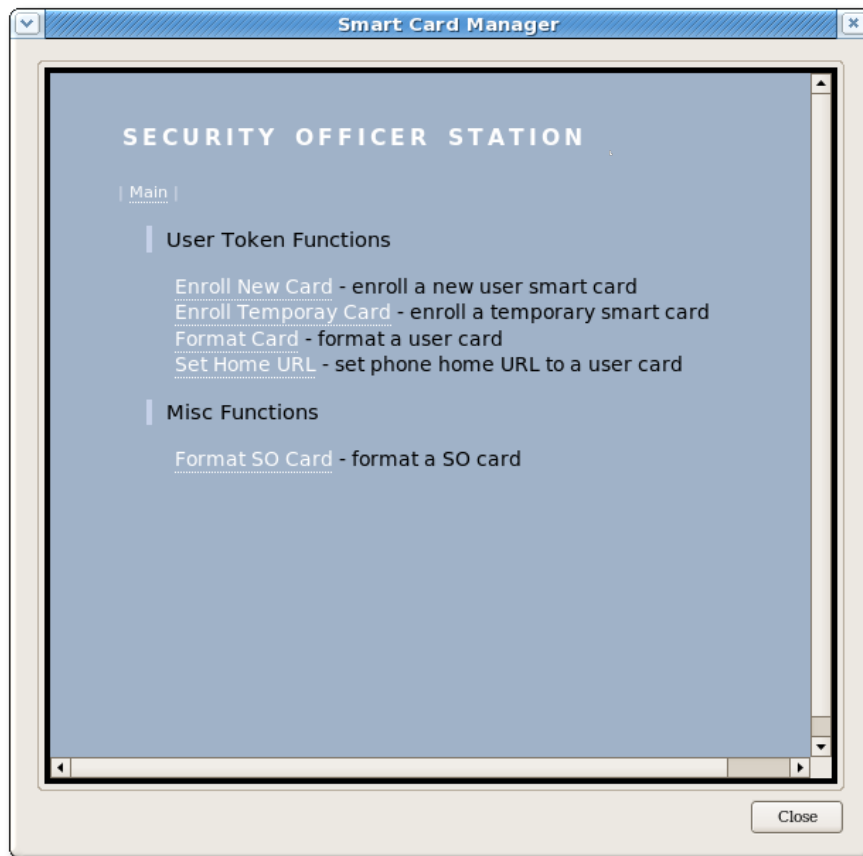


NOTE

Ensure that there is a valid and enrolled security officer card plugged into the computer. A security officer's credentials are required to access the following pages.



3. Click **Continue** to display the security officer Station page. You may be prompted to enter the password for the security officer's card. This is required for SSL client authentication.
4. Select the operation from the menu (enrolling a temporary token, formatting the card, or setting the Phone Home URL).



5. Continue the operation as described in [Chapter 3, Using the Enterprise Security Client](#).

4.3.3. Formatting an Existing Security Officer Smart Card



NOTE

An e-gate token cannot be formatted on a Windows machine using the Enterprise Security Client. To format an e-gate card, perform the operation on Red Hat Enterprise Linux or Mac.



IMPORTANT

Reformatting a token is a destructive operation to the security officer's token and should only be done if absolutely needed.

1. Click **Format SO Card**. Because the security officer card is already inserted, the following screen displays:



2. Click **Format** to begin the operation.

When the card is successfully formatted, the security officer's card values are reset. Another security officer's card must be used to enter security officer mode and perform any further operations.

Chapter 5. Using Keys on Smart Cards for Web and Mail Clients

After a token is enrolled, the token can be used for SSL client authentication and S/MIME email applications. The PKCS #11 module has different names and is located in different directories depending on the operating system.

Table 5.1. PKCS #11 Module Locations

Platform	Module Name	Location
Red Hat Enterprise Linux	libcoolkeypk11.so	/usr/lib/
Windows	coolkeypk11.dll	C:\Windows\System32\
Mac OS X 10.5.8 and higher	libcoolkeypk11.dylib	/Library/Application Support/CoolKey/PKCS11

5.1. Setting up Browsers to Support SSL for Tokens

This is the procedure for Mozilla Firefox. The procedure for setting up certificates for browsers varies for each browser.

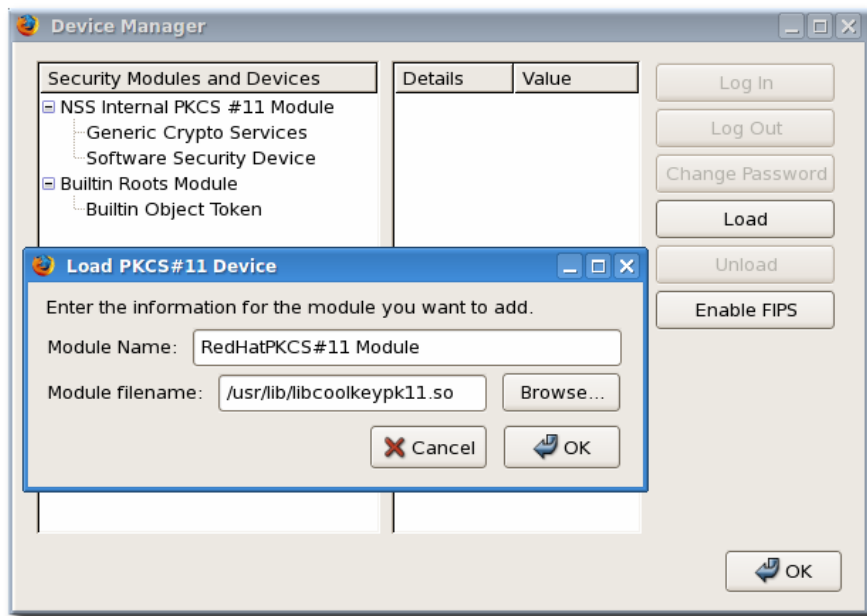
1. In Mozilla Firefox, open the **Tools** menu, choose **Options**, and then click **Advanced**.
2. Add a PKCS #11 driver.



NOTE

Windows and Mac systems automatically attempt to load the PKCS #11 module to any Mozilla browsers they find. Only load the PKCS#11 modules manually if there is a problem detecting the module automatically or if a browser is installed after Enterprise Security Client is installed.

- a. Click **Manage Security Devices** to open the **Device Manager** window, and then click the **Load** button.
- b. Enter a module name, such as **token key pk11 driver**.
- c. Click **Browse**, find the Enterprise Security Client PKCS #11 driver, and click **OK**.



3. If the CA is not yet trusted, download and import the CA certificate.

a. Open the **SSL End Entity** page on the CA. For example:

<https://server.example.com:9444/ca/ee/ca/>

b. Click the **Retrieval** tab, and then click **Import CA Certificate Chain**.

c. Click **Download the CA certificate chain in binary form** and then click **Submit**.

d. Choose a suitable directory to save the certificate chain, and then click **OK**.

e. Click **Edit > Preferences**, and select the **Advanced** tab.

f. Click the **View Certificates** button.

g. Click **Authorities**, and import the CA certificate.

4. Set the certificate trust relationships.

a. Click **Edit > Preferences**, and select the **Advanced** tab.

b. Click the **View Certificates** button.

c. Click **Edit**, and set the trust for websites.

The certificates can be used for SSL.

5.2. Using the Certificates on Tokens for Mail Clients

This is the procedure for Mozilla Thunderbird. The procedure for setting up certificates for mail clients varies for each client.

1. In Mozilla Thunderbird, open the **Edit** menu, and select **Account Settings**.

2. Select **Security** on the left.

3. Add a PKCS #11 driver.

a. Click **Manage Security Devices** to open the **Device Manager** window.

b. Click the **Load** button.

c. Enter the module name, such as **token keypk11 driver**.

d. Click **Browse**, find the Enterprise Security Client PKCS #11 driver, and click **OK**.

4. If the CA is not yet trusted, download and import the CA certificate.
 - a. Open the **SSL End Entity** page on the CA. For example:

https://server.example.com:9444/ca/ee/ca/
 - b. Click the **Retrieval** tab, and then click **Import CA Certificate Chain**.
 - c. Click **Download the CA certificate chain in binary form** and then click **Submit**.
 - d. Choose a suitable directory to save the certificate chain, and then click **OK**.
 - e. In Thunderbird, open the **Edit** menu, and select **Account Settings**.
 - f. Select **Security** on the left, and click the **Manage Certificates** button.
 - g. Click the **Authorities** tab, and import the CA certificate.
5. Set up the certificate trust relationships.
 - a. In Thunderbird, open the **Edit** menu, and select **Account Settings**.
 - b. Select **Security** on the left, and click the **Manage Certificates** button.
 - c. In the **Authorities** tab, select the CA, and click the **Edit** button.
 - d. Set the trust settings for identifying websites and mail users.
 - e. In the **Digital Signing** section of the **Security** panel, click **Select** to choose a certificate to use for signing messages.
6. In the **Encryption** of the **Security** panel, click **Select** to choose the certificate to encrypt and decrypt messages.

5.3. Setting up Mail and Browser Clients on Mac OS X

On Mac, both Safari and Apple Mail are configured to use a smart card for authentication by adding the certificates to the Apple keystore.

1. Enroll a smart card that contains your email address in the certificate, such as [Section 3.5, “Enrolling a Smart Card Automatically”](#).
 For convenience, the TPS server can be configured to use an LDAP directory for user and email information. This is covered in [Section 3.4, “Setting up Users to Be Enrolled”](#).
2. Download the CA certificate.
 - a. Open the **SSL End Entity** page on the CA. For example:

https://server.example.com:9444/ca/ee/ca/
 - b. Click the **Retrieval** tab, and then click **Import CA Certificate Chain**.
 - c. Select the **Display certificates in the CA certificate chain for importing individually into a server** radio button.
 - d. The browser displays a list of certificates in base-64 format. Pick the first blob displayed, and create and save a text file, with a name like **ca.cer**.
3. Import the CA certificate text file using the Apple KeyChain utility, and set the trust options.
 - a. Click on the **System** keychain.
 - b. In the main menu, click **File|Import Items**.
 - c. Browse to the location where the CA certificate file, **ca.cer**, is saved.
 - d. During the import operation, agree to trust the certificate **always**.
4. Insert the enrolled CoolKey token.

5. When the **KeyChain access** utility opens, the new keychain, with your name, is displayed.
6. Locate the certificates under the smart card's keychain.
7. Drag and drop the smart card certificates into the **login** keychain.

Chapter 6. Configuring the Enterprise Security Client

The Enterprise Security Client is now based on Mozilla XULRunner, allowing the preferences facility built into Mozilla to be used for simple configuration of the Enterprise Security Client. A simple UI, discussed in [Chapter 3, Using the Enterprise Security Client](#), manages most important configuration settings.



NOTE

The Enterprise Security Client can be launched without requiring extra configuration.

6.1. Overview of Enterprise Security Client Configuration

The Enterprise Security Client is an intermediary frontend that provides connections between users (and their tokens), the Token Processing System, and certificate authority. The Enterprise Security Client provides two slightly different interfaces:

- ▶ A local interface, based on XUL and JavaScript
- ▶ A web-hosted interface which can be used for remote access, based on CGIs, HTML, and JavaScript

The primary Enterprise Security Client user interface, which is accessed from the local server, incorporates Mozilla XULRunner technology. XULRunner is a runtime package which hosts standalone applications based on XUL, an XML markup language with a rich feature set for user interfaces and offers several advantages over HTML for applications:

- ▶ A wide UI widget set and greater control over the presentation.
- ▶ Local markup to the client machine, so it has a greater privilege level than HTML.
- ▶ JavaScript as the scripting language for convenient program logic scripting and the ability to leverage XPCOM technology.

All of the files for the web-hosted interface can be customized and edited to change the behavior or appearance of the Enterprise Security Client, within reason.

The Enterprise Security Client, in conjunction with the Token Processing System, supports different *user profiles* so that different types of users have different token enrollment paths. Both the Enterprise Security Client and TPS also support different *token profiles*, so that the certificate settings can be custom-defined for different types of tokens. Both of these configurations are set in the TPS, and are described in the *Certificate System Administrator's Guide*.

6.1.1. About the Preferences Configuration Files

The Enterprise Security Client is configured similarly to Mozilla applications, using preferences files. The primary configuration file is **esc-prefs.js**, which is installed with Enterprise Security Client. The second one is **prefs.js** in the Mozilla profiles directory, which is created when the Enterprise Security Client is first launched.

The Enterprise Security Client uses the Mozilla configuration preferences for each of the supported platforms. A default configuration file is located in the following directories on each platform:

- ▶ On Windows, this is in **C:\Program Files\Red Hat\ESC\defaults\preferences\esc-prefs.js**.
- ▶ On Red Hat Enterprise Linux 32-bit, this is in **/usr/lib/esc-1.1.0/defaults/preferences/esc-prefs.js**.

- On Red Hat Enterprise Linux 64-bit, this is in **/usr/lib64/esc-1.1.0/defaults/preferences/esc-prefs.js**.
- On Mac, this is in **~/Desktop/ESC.app/defaults/preferences/esc-prefs.js**.

The **esc-prefs.js** file specifies the default configuration to use when the Enterprise Security Client is first launched. This includes parameters to connect to the TPS subsystem, to use CAPI on Windows, set the password prompt, and configure Phone Home information. Each setting is prefaced by the word **pref**, then the parameter and value are enclosed in parentheses. For example:

```
pref(parameter, value);
```

The **esc-prefs.js** file parameters are listed in [Table 6.1, “esc-prefs.js Parameters”](#). The default **esc-prefs.js** file is shown in [Example 6.1, “Default esc-prefs.js File”](#).

Table 6.1. esc-prefs.js Parameters

Parameter	Description	Notes and Defaults
toolkit.defaultChromeURI	Defines the URL for the Enterprise Security Client to use to contact the XUL Chrome page.	("toolkit.defaultChromeURI", "chrome://esc/content/settings.xul")
esc.tps.message.timeout	Sets a timeout period, in seconds, for connecting to the TPS.	("esc.tps.message.timeout", "90");
esc.windows.do.capi	Enables the client to connect to Windows CAPI. This writes newly-created certificates to the local CAPI store after an enrollment operation and removes those certificates when the formatting operation is complete.	("esc.windows.do.capi", "yes");
esc.disable.password.prompt	Enables the password prompt, which means that a password is required to read the certificate information off the smart card. The password prompt is disabled by default, so anyone can use the Enterprise Security Client. However, in security contexts, like when a company uses security officers to manage token operations, then this should be enabled, to restrict access to the Enterprise Security Client.	("esc.disable.password.prompt", "yes");
esc.global.phone.home.url	Sets the URL to use to contact the TPS server. Normally, the Phone Home information is set on the token already through its applet. If a token does not have Phone Home information, meaning it has no way to contact the TPS server, then the Enterprise Security Client checks for a global default Phone Home URL. This setting is only checked if it is explicitly set. This setting also applies to every token formatted through the client, so setting this parameter forces all tokens to point to the same TPS. Only use this parameter if that specific behavior is desired.	("esc.global.phone.home.url", "http://server.example.com:7888/cgi-bin/home/index.cgi");
esc.global.alt.nss.db	Points to a directory that contains a common security database that is used by all Enterprise Security Client users on the server. Phone Home URL.	prefs("esc.global.alt.nss.db", "C:/Documents and Settings/All Users/shared-db");

This setting is only checked if it is explicitly set. If this is not set, then each user accesses only each individual profile security database, rather than a shared database.

Example 6.1. Default esc-prefs.js File

The comments in this file are not included in the example.

```
#pref("toolkit.defaultChromeURI", "chrome://esc/content/settings.xul");
pref("signed.applets.codebase_principal_support", true); for internal use only

pref("capability.principal.codebase.p0.granted", "UniversalXPConnect"); for
internal use only
pref("capability.principal.codebase.p0.id", "file:///"); for internal use only

pref("esc.tps.message.timeout", "90");

#Do we populate CAPI certs on windows?
pref("esc.windows.do.capi", "yes");

#Sample Security Officer Enrollment UI
#pref("esc.security.url", "http://test.host.com:7888/cgi-bin/so/enroll.cgi");

#Sample Security Officer Workstation UI
#pref("esc.security.url", "https://dhcp-170.sjc.redhat.com:7889/cgi-
bin/sow/welcome.cgi");

#Hide the format button or not.
pref("esc.hide.format", "no");

#Use this if you absolutely want a global phone home url for all tokens
#Not recommended!
#pref("esc.global.phone.home.url", "http://test.host.com:7888/cgi-
bin/home/index.cgi");
```

When the Enterprise Security Client is launched, it creates a separate, unique profile directory for each user on the system. These profiles are stored in different locations on each platform:

- ▶ On Windows, this is in **C:\Documents and Settings\%USER%\Application Data\RedHat\ESC\Profiles\alphanumeric_string.default\prefs.js**.
- ▶ On Red Hat Enterprise Linux, this is in **~/.redhat/esc/alphanumeric_string.default/prefs.js**.
- ▶ On Mac, this is in **~/Library/Application Support/ESC/Profiles**.



NOTE

When the Enterprise Security Client requires any changes to a user's configuration values, the updated values are written to the user's profile area, not to the default JavaScript file.

[Table 6.2, “prefs.js Parameters”](#) lists the most relevant parameters for the **prefs.js** file. Editing this file

is tricky. The **prefs.js** file is generated and edited dynamically by the Enterprise Security Client, and manual changes to this file are overwritten when the Enterprise Security Client exits.

Table 6.2. prefs.js Parameters

Parameter	Description	Notes and Defaults
esc.tps.url	Sets a URL for the Enterprise Security Client to use to connect to the TPS. This is not set by default.	
esc.key. <i>token_ID</i> .tps.url	Sets the hostname and port to use to contact a TPS. If this Phone Home information was not burned into the card at the factory, it can be manually added to the card by adding the TPS URL, an enrollment page URL, the issuer's name, and Phone Home URL.	("esc.key. <i>token_ID</i> .tps.url" = "http://server.example.com:7888/nk_service");
esc.key. <i>token_ID</i> .tps.enrollment-ui.url	Gives the URL to contact the enrollment page for enroll certificates on the token. If this Phone Home information was not burned into the card at the factory, it can be manually added to the card by adding the TPS URL, an enrollment page URL, the issuer's name, and Phone Home URL.	("esc.key. <i>token_ID</i> .tps.enrollment-ui.url" = "http://server.example.com:7888/cgi_bin/esc.cgi?");
esc.key. <i>token_ID</i> .issuer.name	Gives the name of the organization enrolling the token.	("esc.key. <i>token_ID</i> .issuer.name" = "Example Corp");
esc.key. <i>token_ID</i> .phone.home.url	Gives the URL to use to contact the Phone Home functionality for the TPS. The global Phone Home parameter sets a default to use with any token enrollment, if the token does not specify the Phone Home information. By setting this parameter to a specific token ID number, the specified Phone Home parameter applies only to that token.	("esc.key. <i>token_ID</i> .phone.home.url" = "http://server.example.com:7888/cgi-bin/home/index.cgi?");
esc.security.url	Points to the URL to use for security officer mode. If this is pointed to the security officer enrollment form, then the Enterprise Security Client opens the forms to enroll security officer tokens. If this is pointed to the security officer workstation URL, then it opens the workstation to enroll regular users with security officer approval.	("esc.security.url","https://server.example.com:7888/cgi-bin/so/enroll.cgi");

6.1.2. About the XUL and JavaScript Files in the Enterprise Security Client

Smart Card Manager stores the XUL markup and JavaScript functionality in `/usr/lib[64]/esc-1.1.0/chrome/content/esc/`.

The primary Enterprise Security Client XUL files are listed in [Table 6.3, “Main XUL Files”](#).

Table 6.3. Main XUL Files

Filename	Purpose
settings.xul	Contains the code for the Settings page.
esc.xul	Contains the code for the Enrollment page.
config.xul	Contains the code for the configuration UI.

The primary **Smart Card Manager** JavaScript files are listed in the following table.

Table 6.4. Main JavaScript Files

Filename	Purpose
ESC.js	Contains most of the Smart Card Manager JavaScript functionality.
TRAY.js	Contains the tray icon functionality.
AdvancedInfo.js	Contains the code for the Diagnostics feature.
GenericAuth.js	Contains the code for the authentication prompt. This prompt is configurable from the TPS server, which requires dynamic processing by the Smart Card Manager .

6.1.3. Enterprise Security Client File Locations

This reference shows the different directories and file locations for the different client machines.

The location of the Enterprise Security Client main directory on the different client platforms is as follows:

Table 6.5. Main Directories for the Enterprise Security Client

Platform	Location
Windows	C:\Program Files\Red Hat\ESC
Red Hat Enterprise Linux	/usr/lib/esc-1.1.0/esc
	/usr/lib64/esc-1.1.0/esc
Mac	/Applications, but users can drag the ESC.app directory anywhere

On Windows, Enterprise Security Client uses the following directories and files:

Table 6.6. Enterprise Security Client File and Directory Locations on Windows

File or Directory	Purpose
C:\Program Files\Red Hat\ESC	Main directory.
application.ini	XULRunner application configuration file
components\	XPCOM components directory.
chrome\	Directory for Chrome components and additional application files for Enterprise Security Client XUL and JavaScript.
defaults\	Enterprise Security Client default preferences.
esc.exe	The executable which launches Enterprise Security Client in XULRunner.
xulrunner\	Privately-deployed XULRunner bundle.

On Red Hat Enterprise Linux 32-bit, the Enterprise Security Client is installed by its binary RPM to the default location, `/usr/lib/esc-1.1.0/esc`. On Red Hat Enterprise Linux 64-bit systems, the installation directory is `/usr/lib64/esc-1.1.0/esc`.

**NOTE**

Unlike Windows and Mac versions of Enterprise Security Client, the Enterprise Security Client on Red Hat Enterprise Linux uses the system XULRunner packages rather than a privately-deployed XULRunner bundle.

Table 6.7. Enterprise Security Client File and Directory Locations on Red Hat Enterprise Linux

File or Directory	Purpose
application.ini	XULRunner application configuration file.
components/	XPCOM components.
chrome/	Directory for Chrome components and additional application files for Enterprise Security Client XUL and JavaScript.
defaults/	Enterprise Security Client default preferences.
esc	The script which launches the Enterprise Security Client.

On Mac OS X, the XULRunner framework located in **ESC.app**.

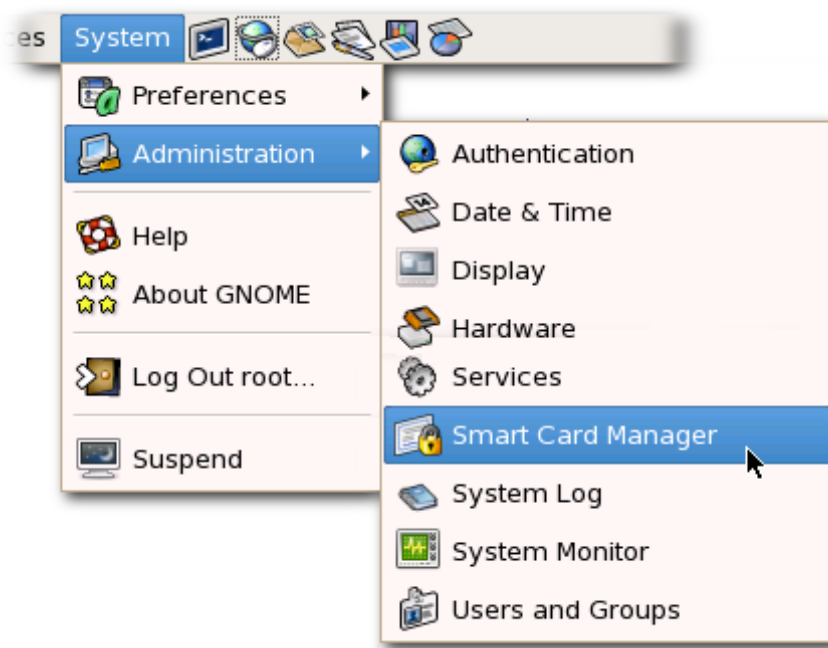
Table 6.8. Enterprise Security Client File and Directory Locations on Mac

File or Directory	Purpose
Contents/	Privately deployed XUL framework ► Info.plist ► Frameworks/ ► XUL.framework/ ► Resources
application.ini	Enterprise Security Client XULRunner application configuration file.
components/	Enterprise Security Client XPCOM components.
chrome/	Directory for Chrome components and additional application files for Enterprise Security Client XUL and Javascript.
defaults/	Enterprise Security Client default preferences.
xulrunner	The script which launches Enterprise Security Client.

6.2. Configuring SSL Connections with the TPS

By default, the TPS communicates with the Enterprise Security Client over standard HTTP. It may be desirable to secure the TPS-client communications by using HTTP over SSL (HTTPS).

1. The Enterprise Security Client has to have the CA certificate for the CA which issued the TPS's certificates in order to trust the TPS connection. Import the CA certificate as described in [Section 3.6.4, "Importing CA Certificates"](#).
2. The Enterprise Security Client needs to be configured to communicate with the TPS over SSL; this is done by setting the *Phone Home URL*, which is the default URL the Enterprise Security Client uses to connect to the TPS.
3. Open the Enterprise Security Client.



4. Insert a new, blank token into the machine.

Blank tokens are unformatted, so they do not have an existing Phone Home URL, and the URL must be set manually. Formatted tokens (tokens can be formatted by the manufacturer or by your IT department) already have the URL set, and thus do not prompt to set the Phone Home URL.

5. Fill in the new TPS URL with the SSL port information. For example:

```
https://server.example.com:7890/cgi-bin/home/index.cgi
```

6. Click the **Test** button to send a message to the TPS.
7. If the request is successful, the client opens a dialog box saying that the Phone Home URL was successfully obtained.

6.3. Using Shared Security Databases

The Enterprise Security Client usually creates a new NSS security database for keys and certificates for each user profile associated with the Enterprise Security Client. Whenever a user imports or trusts a certificate for the Enterprise Security Client to use, it is imported into that NSS database for that profile. (This is similar to the way that web browsers have different user profiles with different security databases, password stores, and bookmarks for each profile.)

There may be instances when there are multiple Enterprise Security Client users who all use the client on a single machine. In that case, it makes sense to have a common, shared security database that is trusted by the Enterprise Security Client in addition to the user profile databases. That shared security database contains certificates that are held in common by all users, such as the CA signing certificate used by the TPS.

Using a shared security database is not configured by default.

1. Stop the Enterprise Security Client.
2. Create the security database directory and the databases that will be shared. Before configuring the Enterprise Security Client, the databases must exist, be readable by the client, and contain the certificates that will be used by the client.

NSS databases can be created using the **certutil** command. See the **certutil** documentation, such as <http://www.mozilla.org/projects/security/pki/nss/tools/certutil.html>, for more information.

3. Open the **esc-prefs.js** file.

```
vim /usr/lib/esc-1.1.0/defaults/preferences/esc-prefs.js
```

4. Add the **esc.global.alt.nss.db** parameter, pointing to the directory which contains the shared database.

```
prefs("esc.global.alt.nss.db", "C:/Documents and Settings/All  
Users/common_db");
```

5. When the Enterprise Security Client is restarted, the configuration changes will be applied.

6.4. Customizing the Smart Card Enrollment User Interface

The TPS subsystem displays a generically-formatted smart card enrollment screen which is opened automatically when an uninitialized smart card is inserted. This is actually comprised of three pages, depending on the mode in which the client is running:

- `/var/lib/pki-tps/cgi-bin/home/Enroll.html` for regular enrollments
- `/var/lib/pki-tps/cgi-bin/so/Enroll.html` for security officer enrollments
- `/var/lib/pki-tps/cgi-bin/sow/Enroll.html` for security officer workstation enrollments (users enrolled through the security officer UI)



NOTE

The security officer workstation directory contains other HTML files for other token operations, such as formats and PIN resets.

There may be even more enrollment pages if there are custom user profiles.

These enrollment pages are basic HTML and JavaScript, which allows them to be easily customized for both their appearance and functionality. The resources, such as images and JavaScript files, referenced by the enrollment file are located in the corresponding **docroot/** directory, such as `/var/lib/pki-tps/docroot/esc/sow` for the security officer enrollment file in `/var/lib/pki-tps/cgi-bin/sow`.

There are several ways that the smart card enrollment pages can be customized. The first, and simplest, is changing the text on the page. The page title, section headings, field names, and descriptions can all be changed by editing the HTML file, as shown in the extracts in [Example 6.2, “Changing Page Text”](#).

Example 6.2. Changing Page Text

```
<!-- Change the title if desired -->
<title>Enrollment</title>
...
<p class="headerText">Smartcard Enrollment</p>
...
<!-- Insert customized descriptive text here. -->
<p class="bodyText">You have plugged in your smart card!
    After answering a few easy questions, you will be able to use your
    smart card.
</p>
<p class="bodyText">
    Now we would like you to identify yourself.
</p>
...
<table>
    <tr>
        <td><p>LDAP User ID: </p></td>
        <td></td>
        <td><input type="text" id="sname" value=""></td>
    </tr>
</table>
```

The styles of the page can be changed through two files: the **style.css** CSS style sheet and the logo image, **logo.png**.

Example 6.3. Changing Page Styles

```
<link rel=stylesheet href="/esc/home/style.css" type="text/css">
...

<table width="100%" class="logobar">
  <tr>
    <td>
      
    </td>
    <td>
      <p class="headerText">Smartcard Enrollment</p>
    </td>
  </tr>
</table>
```

The **style.css** file is a standard CSS file, so all of the tags and classes can be defined as follows:

```
body {
background-color: grey;
    font-family: arial;
    font-size: 7p
}
```

More information on CSS is available at <http://www.w3.org/Style/CSS/learning>.

The last way to customize the **Enroll.html** files is through the JavaScript file which sets the page functionality. This file controls features like the progress meter, as well as processing the inputs which are used to authenticate the user to the user directory.

Example 6.4. Changing Page Script

```
<progressmeter id="progress-id" hidden="true" align = "center"/>
...
<table>
  <tr>
    <td><p >LDAP User ID: </p></td>
    <td> </td>
    <td><input type="text" id="sname" value=""></td>
  </tr>
</table>
```

**WARNING**

Be very cautious about changing the **util.js** file. If this file is improperly edited, it can break the Enterprise Security Client UI and prevent tokens from being enrolled.

The complete **/var/lib/pki-tps/cgi-bin/home/Enroll.html** file is in [Example 6.5, "Complete Enroll.html File"](#).

Example 6.5. Complete Enroll.html File

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<link rel=stylesheet href="/esc/home/style.css" type="text/css">

<title>Enrollment</title>
</head>
<script type="text/JavaScript" src="/esc/home/util.js">
</script>
<body onload="InitializeBindingTable();" onunload=cleanup()>

<progressmeter id="progress-id" hidden="true" align = "center"/>
<table width="100%" class="logobar">
  <tr>
    <td>

    </td>
    <td>
      <p class="headerText">Smartcard Enrollment</p>
    </td>
  </tr>
</table>
  <table id="BindingTable" width="200px"align="center">
    <tr id="HeaderRow">
    </tr>
  </table>
  <p class="bodyText">You have plugged in your smart card! After answering a few
easy questions, you will be able to use your smart card.
</p>
  <p class="bodyText">
    Now we would like you to identify yourself.
  </p>
  <table>
    <tr>
      <td><p >LDAP User ID: </p></td>
      <td> </td>
      <td><input type="text" id="snameuf" value=""></td>
      <td> </td>
      <td><p>LDAP Password: </p></td>
      <td> </td>
      <td><input type="password" id="snamepwd" value=""></td>
    </tr>

  </table>

  <p class="bodyText"> Before you can use your smart card, you will need a
password to protect it.</p>
  <table>
    <tr>
      <td><p >Password:</p></td>
      <td><input type="password" id="pintf" name="pintf" value=""></td>

      <td><p >Re-Enter Password:</p></td>
      <td><input type="password" id="reenterpintf" name="reenterpintf"
value=""></td>
    </tr>
  </table>
  <br>
  <table width="100%">
    <tr>
      <td align="right">

```

```

        <input type="button" id="enrollbtn" name="enrollbtn" value="Enroll My
Smartcard" onClick="DoEnrollC00LKey();">
    </td>
</tr>
</table>
</body></html>

```

6.5. Disabling LDAP Authentication for Token Operations

By default, each user who requests a token operation is authenticated against an LDAP directory. If the user has an entry, then the operation is allowed; if the user does not have an entry, then the operation is rejected.

For testing or for certain types of users, then it may be simpler or preferable to disable LDAP authentication. This is not configured in the Enterprise Security Client configuration, but in the Token Processing System configuration, and must be done by a TPS administrator.

1. Stop the TPS subsystem.

```
service pki-tps stop
```

2. Set the authentication parameters to **false**.

```

op.operation_type.token_type.loginRequest.enable=false
op.operation_type.token_type.auth.enable=false

```

The *operation_type* is the token operation for which LDAP authentication is being disabled, such as **enroll**, **format**, or **pinreset**. Disabling authentication for one operation type does not disable it for any other operation types.

The *token_type* is the token profile. There are default profiles for regular users, security officers, and the users enrolled by security officers. There can also be custom token types for other kinds of users or certificates.

For example:

```

op.enroll.userKey.loginRequest.enable=false
op.enroll.userKey.pinReset.enable=false

```

3. Restart the TPS subsystem.

```
service pki-tps start
```

Editing the TPS configuration is covered in the *Certificate System Administrator's Guide*.